



HAL
open science

Intrinsically Motivated Goal Exploration in Child Development and Artificial Intelligence: Learning and Development of Speech and Tool Use

Sébastien Forestier

► **To cite this version:**

Sébastien Forestier. Intrinsically Motivated Goal Exploration in Child Development and Artificial Intelligence: Learning and Development of Speech and Tool Use. Artificial Intelligence [cs.AI]. Université de Bordeaux, 2019. English. NNT : 2019BORD0247 . tel-03438828

HAL Id: tel-03438828

<https://theses.hal.science/tel-03438828v1>

Submitted on 22 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE
**DOCTEUR DE
L'UNIVERSITÉ DE BORDEAUX**

ÉCOLE DOCTORALE DE
MATHÉMATIQUES ET INFORMATIQUE
SPÉCIALITÉ : INFORMATIQUE

Par Sébastien Forestier

**Intrinsically Motivated Goal Exploration in
Child Development and Artificial Intelligence:
Learning and Development of Speech and Tool Use**

Sous la direction de Pierre-Yves Oudeyer

Date de soutenance : 22 Novembre 2019

Président du jury :

Jochen Triesch, Professor, Frankfurt Institute for Advanced Studies

Rapporteurs :

Verena Hafner, Professor of Adaptive Systems, Humboldt-Universität zu Berlin

Justus Piater, Professor of Computer Science, Universität Innsbruck

Examineurs :

Nivedita Mani, Professor of Psychology of Language, Universität Göttingen

Frank Guerin, Lecturer, Department of Computing Science, University of Aberdeen

Directeur de thèse :

Pierre-Yves Oudeyer, Directeur de recherche, INRIA Bordeaux

Titre : Exploration intrinsèquement motivée orientée vers des buts dans le développement de l'enfant et en intelligence artificielle : apprentissage et développement de la parole et de l'utilisation des outils.

Résumé : Les bébés et enfants humains sont curieux, ils explorent activement leur monde. Un de leurs premiers défis est l'apprentissage des relations de causalité entre leurs actions, telles que les mouvements de leurs bras ou leur voix, et les changements dans l'environnement. Les motivations intrinsèques pourraient être un des mécanismes clés de cet apprentissage, mais elles ont été peu étudiées en psychologie du développement. Par ailleurs, les robots qui apprennent aujourd'hui des compétences avancées le font d'une manière très différente de celle des enfants humains. Cette thèse présente deux objectifs complémentaires : d'une part la compréhension du rôle des motivations intrinsèques dans le développement de la parole et de l'utilisation d'outils chez l'enfant à travers la modélisation robotique, et d'autre part l'amélioration des capacités des robots à apprendre ces compétences par l'implémentation de mécanismes inspirés par l'apprentissage des enfants. La première partie de ce travail concerne la compréhension et modélisation des motivations intrinsèques chez l'humain. Nous réanalysons une expérience d'évaluation des capacités d'utilisation d'outils par les enfants, et montrons que les motivations intrinsèques semblent jouer un rôle important dans les comportements observés et même interférer avec les mesures de succès dans la tâche. Avec un modèle robotique, nous montrons qu'une motivation intrinsèque basée sur le progrès pour atteindre ses propres buts, couplée à une représentation modulaire de ces buts peut auto-organiser des phases de comportements dans le développement des précurseurs de l'utilisation d'outils qui ont des propriétés similaires avec ce développement chez les enfants. Nous présentons le premier modèle robotique de l'apprentissage de la parole et de l'utilisation d'outils à partir de zéro, qui permet de prédire que l'exploration d'objets physiques dans un scénario d'interaction sociale accélère l'apprentissage de la vocalisation de sons pour nommer ces objets en conséquence d'une exploration des objets orientée vers des buts. Dans la seconde partie de cette thèse, nous développons, formalisons et évaluons ces algorithmes avec pour but d'obtenir un apprentissage robotique efficace. Nous formalisons une approche algorithmique appelée Intrinsically Motivated Goal Exploration Processes (IMGEP), qui permet la découverte et l'acquisition d'un vaste répertoire de compétences grâce aux motivations intrinsèques. Nous démontrons dans différents environnements robotiques dont un comprenant un robot humanoïde que l'apprentissage de divers espaces de buts avec des motivations intrinsèques est plus efficace pour l'apprentissage de compétences complexes que de seulement s'intéresser à l'apprentissage de ces compétences.

Mots-clés : Motivations intrinsèques ; Exploration orientée vers des buts ; Développement de l'enfant ; Intelligence artificielle ; Robotique développementale ; Parole ; Utilisation d'outils.

Title: Intrinsically motivated goal exploration in child development and artificial intelligence: learning and development of speech and tool use.

Summary: Babies and children are curious, active explorers of their world. One of their first challenges is to learn the relations between their actions, such as the use of tools or speech, and the changes in their environment. Intrinsic motivations could be a key mechanism of this learning, but they have been little studied in developmental psychology. Also, robots that learn advanced skills today learn in a way very different from human children. The objective of this thesis is twofold: understanding the role of intrinsic motivations in human development of speech and tool use through robotic modeling, and improving the abilities of artificial agents inspired by the mechanisms of human exploration and learning. A first part of this work concerns the understanding and modeling of intrinsic motivations. We reanalyze a typical tool-use experiment and show that intrinsically motivated exploration seems to play an important role in the observed behaviors and to interfere with the measured success rates. With a robotic model, we show that an intrinsic motivation based on the progress to reach goals with a modular representation can self-organize phases of behaviors in the development of tool-use precursors that share properties with child development. We present the first robotic model learning both speech and tool use from scratch, which predicts that the grounded exploration of objects in a social interaction scenario accelerates infant vocal learning of sounds to name these objects as a result of a goal-directed exploration of objects. In the second part of this thesis, we extend, formalize and evaluate the algorithms designed to model child development, with the aim to build an efficient learning robot. We formalize an approach called Intrinsically Motivated Goal Exploration Processes (IMGEP) that enables the discovery and acquisition of large repertoires of skills through intrinsic motivations. We show within several experimental setups including a real humanoid robot that learning diverse spaces of goals with intrinsic motivations is more efficient for learning complex skills than only trying to learn these skills.

Keywords: Intrinsic motivations; Goal exploration; Child development; Artificial intelligence; Developmental robotics; Speech; Tool use.

Unité de recherche : Équipe-projet FLOWERS, Inria Bordeaux Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence, France.

Parmi les pionniers de ce mouvement, deux personnages restent selon moi à jamais inoubliables : Fridtjof Nansen pour l'Arctique, et Ernest Shackleton pour l'Antarctique. Parce que l'un et l'autre, malgré les différences qui les séparent, sont toujours restés d'abord des hommes, qui plaçaient au second rang leur ambition d'être les premiers dans ces fameuses "courses" qui tant fascinaient leurs contemporains. Ni l'un ni l'autre n'atteignirent jamais le but ultime : le Pôle. [...]
Et l'échec apparent d'une entreprise peut générer plus de résultats que bien des prétendus succès. Car telle est la vertu première de l'aventurier digne de ce beau nom : faire que l'aventure soit déjà en elle-même par-delà les aléas du meilleur et du pire, du succès et de l'échec, un accomplissement.

PAUL-ÉMILE VICTOR
préface de "L'Odyssée de l'Endurance"
de Sir Ernest Shackleton

Acknowledgements

I would like to express my deepest gratitude to my advisor Pierre-Yves Oudeyer. This adventure would not have looked the same without your invaluable scientific guidance and your continuous support.

I'm extremely grateful to the members of the jury Verena Hafner, Nivedita Mani, Justus Piater, Frank Guerin and Jochen Triesch, who reviewed this manuscript, attended my defense in Bordeaux and provided valuable and constructive feedback. Jochen, thank you for welcoming me in your lab as an intern and offering me an insightful experience a while before my PhD.

I would like to extend my sincere thanks to René Adad, mathematics teacher in Lycée Thiers, and Luc Bougé, head of the Computer Science department at ENS Rennes, for their unparalleled ability to communicate their passion and enthusiasm for their scientific domain.

During those four years, it was a great pleasure and rich experience to belong to the Flowers team at INRIA. I have to thank all the people I had the chance to meet, for sharing their time to discuss, learn and explore on both scientific and human sides. Special mention is due to the engineers of the team, Yoan Mollard, Théo Segonds and many others, who were instrumental to the success of the robotic experiments. I am also grateful to my interns Timothée Anne, Alexandre Péré and Rémy Portelas who accepted the challenge of working on risky topics with me. Thanks to Baptiste for always discussing a diversity of random topics, one of which inspired a passion in me. Rémy you have been a great partner in our different endeavours, and I have to confess some experts think the student have surpassed the master. I would like to also thank for their support Jérôme, Jonathan, Haylee, Mai, Fabien, Pierre, Clément, William, Adrien, Cédric, Octave, Tallulah, and all the others I cannot cite due to space. Arthur, although far away doing your own PhD, I feel you have been part of my team.

I was lucky to collaborate with psychologists studying the development of babies. Many thanks to Lauriane Rat-Fischer who was happy to share her data and analyze it with us. I hope this collaboration was as insightful for you experimenting with babies as it was for us doing robotic models. Thanks also to Lisa Jacquy, for your lively interest in our work and for our discussions on the many definitions of goals and curiosity.

Je voudrais remercier ma famille pour avoir supporté mon éloignement lors de ces longues études. Merci à mes parents, vous m'avez apporté tout votre soutien quand j'en ai eu besoin. Ces accomplissements vous doivent beaucoup.

Por fin gracias a ti, has sido la tienda luz de este viaje.

Résumé

Les bébés et enfants humains sont curieux, ils explorent activement leur monde. Ils jouent avec les objets et les personnes présentes dans leur environnement, sans qu'on le leur demande, et potentiellement jusqu'à ce qu'on les arrête ou qu'ils s'endorment. Cette motivation intrinsèque pour explorer et apprendre de nouvelles compétences et connaissances est présente dans toutes leurs situations d'apprentissage et pourrait être un des mécanismes importants du développement. Cependant, les mécanismes de la curiosité et des motivations intrinsèques ont été peu étudiées en psychologie du développement, et les motivations intrinsèques sont souvent négligées dans l'interprétation des résultats des expériences. Les mécanismes particuliers des motivations intrinsèques sont méconnus, comme la façon de sélectionner des buts et stratégies ou comment le guidage social interagit avec cette sélection autonome.

Par ailleurs, la plupart des agents artificiels et robots apprennent d'une manière très différente de celle des enfants humains. Certains nécessitent une énorme base de donnée d'apprentissage et cherchent des schémas statistiques d'une façon passive, d'autres ont besoin d'un apport de connaissances d'un expert humain pour lancer et guider l'apprentissage, et beaucoup requièrent des millions d'itérations d'apprentissage là où l'enfant n'a besoin que de quelques jours ou semaines. Modéliser les motivations intrinsèques des enfants en les implémentant chez des robots pourrait nous permettre de mieux comprendre leurs mécanismes. D'autre part, concevoir des robots qui apprennent et se développent comme des humains, à travers l'exploration autonome et l'apprentissage de compétences variées pourrait améliorer les capacités de ces robots, comme la vitesse, la robustesse et l'adaptabilité de leur apprentissage, en particulier quand le guidage humain n'est pas disponible.

Un des défis rencontrés par les bébés dès leurs premiers mois est l'apprentissage des relations de causalité entre leurs actions et les changements dans l'environnement. Cet apprentissage prend place d'une manière active, quand le bébé expérimente des actions, qui semblent maladroitement, et observe leurs résultats à la manière d'un petit scientifique. Cela peut être des mouvements des muscles de tous ses membres, ou bien des articulations de son pharynx et larynx, ce qui produit des sons, et leurs résultats peuvent être variés et dans toute modalité sensorielle. L'apprentissage des relations entre ses actions et leurs réactions sur les objets et personnes de son environnement implique une expérimentation active et une découverte des causalités entre ces objets et perceptions sensorielles. L'émergence de l'utilisation des outils et du langage au cours des premières années de vie est en grande partie un mystère, et les possibles liens entre ces deux compétences ne sont pas élucidés. Le concept de motivations intrinsèques a peu été considéré pour rendre compte du développement

de ces compétences, bien que les celles-ci semblent y jouer un rôle important. Le développement de l'utilisation d'outils et de la parole chez les enfants ainsi que leur modélisation chez les robots semblent donc être un terrain d'étude intéressant pour mieux comprendre les motivations intrinsèques.

Cette thèse présente deux objectifs complémentaires : d'une part la compréhension du rôle des motivations intrinsèques dans le développement de la parole et de l'utilisation d'outils chez l'enfant à travers la modélisation robotique, et d'autre part l'amélioration des capacités des robots à apprendre à parler et à utiliser des outils grâce à une inspiration par les mécanismes d'exploration et d'apprentissage humains.

La première partie de ce travail concerne donc la compréhension et modélisation des motivations intrinsèques chez l'humain. Les expériences de psychologie du développement s'attachent pour beaucoup à évaluer des compétences particulières chez les enfants de tout âge, par la mise en place d'une tâche expérimentale que l'enfant est encouragé à résoudre. Cependant, les enfants peuvent aussi suivre leurs propres motivations pour explorer l'appareil expérimental ou d'autres éléments de l'environnement. Nous suggérons que considérer les possibles motivations intrinsèques des enfants dans ces expériences peut aider à la compréhension de leur rôle dans l'apprentissage des compétences associées ainsi que dans le développement à long terme de l'enfant de manière générale. Pour illustrer cette idée, nous réanalysons et réinterprétons une expérience d'évaluation des capacités d'utilisation d'outils par les enfants autour de deux ans, dont la mise en place est typique de ce genre d'expériences. Nous montrons que les motivations des enfants dans cette tâche sont très diverses et ne coïncident souvent pas avec l'objectif attendu et souligné par l'expérimentateur. Les motivations intrinsèques semblent jouer un rôle important dans les comportements observés et même interférer avec les mesures de succès dans la tâche. Cependant, notre analyse a ses propres limites et il serait intéressant d'étudier les comportements intrinsèquement motivés dans une expérience dédiée, avec des buts possibles ainsi que des stratégies pour les résoudre aussi variés que possible, avec un nombre d'expériences plus élevé, ainsi que des moyens d'analyse automatiques des comportements à observer.

Dans le but de modéliser certains aspects du développement de l'utilisation d'outils dans les premières années, nous implémentons ensuite un agent artificiel intrinsèquement motivé, qui génère de lui-même ses propres buts et les sélectionne grâce à des récompenses intrinsèques, le tout dans un environnement simulé en 2D où un bras robotique peut interagir avec différents objets. Avec ce modèle, nous étudions comment des implémentations particulières des motivations intrinsèques pour générer des buts intéressants ainsi qu'une représentation particulière de ces buts peuvent jouer un rôle dans une progression de l'utilisation d'outils. Nous montrons qu'une motivation intrinsèque basée sur le progrès pour atteindre ses propres buts, couplée à une représentation modulaire de ces buts peut auto-organiser des phases de comportements dans le développement des précurseurs de l'utilisation d'outils qui

ont des propriétés en commun avec ce développement observé chez les enfants. Cette implémentation des motivations intrinsèques est compatible avec les observations dans les expériences typiques d'évaluation des compétences avec les outils chez les jeunes enfants, et pourtant les motivations intrinsèques sont souvent négligées dans l'interprétation de ces expériences.

Certaines études ont proposé que le développement de l'utilisation d'outils et celui de la parole ont certains points communs dans les premières années chez les enfants et pourraient avoir des liens dans leurs aspects cognitifs. Pour étudier les mécanismes sous-jacents, nous présentons le premier modèle robotique de l'apprentissage de la parole et de l'utilisation d'outils à partir de zéro. Ce modèle ne présuppose pas de capacités pour le séquençage d'actions complexes ou la planification combinatoire, qui sont pourtant souvent considérés comme nécessaires au développement de ces compétences. Même sans ces capacités, notre modèle robotique peut découvrir progressivement comment attraper des objets avec la main, utiliser des objets comme des outils pour atteindre encore d'autres objets, produire des sons avec sa voix, et faire de ces sons un outil social pour utiliser le parent pour atteindre des objets autrement inatteignables. La découverte que certains sons peuvent être utilisés comme un outil social peut guider l'apprentissage vocal davantage. Ce modèle prédit que l'exploration des objets physiques dans un scénario d'interaction sociale accélère l'apprentissage de la vocalisation des noms de ces objets en conséquence d'une exploration des objets orientée vers des buts. Cependant, ce modèle présuppose l'existence de mécanismes perceptuels déjà développés, bien que dans le développement initial de ces capacités, la perception s'améliore continûment. Cette modélisation bénéficierait par ailleurs d'une comparaison plus directe et précise avec des observations expérimentales pour pouvoir affiner ses mécanismes.

Dans la seconde partie de cette thèse, nous développons, formalisons et évaluons les algorithmes définis pour la modélisation du développement de l'enfant, avec pour but d'obtenir un apprentissage robotique efficace, qui requiert peu de connaissances expertes de la part de l'utilisateur, et qui peut s'adapter à de nouvelles situations d'apprentissage dans une perspective d'apprentissage ouvert. Nous considérons en particulier les architectures d'apprentissage orienté vers des buts qui ont précédemment été développées pour l'exploration et l'apprentissage de solutions à des champs de problèmes sensorimoteurs. Ces architectures n'ont cependant pas été utilisées jusqu'à présent pour l'apprentissage dans des espaces d'effets continus de grande dimension. Nous montrons les limites des architectures existantes pour l'exploration de tels espaces et introduisons une nouvelle architecture appelée Model Babbling (MB). MB exploite efficacement une représentation modulaire de l'espace d'effets, et une version active de MB (AMB) améliore cet apprentissage davantage. Ces architectures sont comparées dans un environnement expérimental simulé avec un bras robotique qui peut découvrir et apprendre comment contrôler des objets en utilisant divers outils, ce qui représente des espaces moteurs et sensoriels continus structurés et de grande dimension.

Nous formalisons ensuite une approche algorithmique appelée *Intrinsically Motivated Goal Exploration Processes* (IMGEP), qui permet la découverte et l'acquisition d'un vaste répertoire de compétences à travers l'auto-génération, auto-sélection, auto-ordonnancement et auto-expérimentation des buts d'apprentissage. L'architecture algorithmique IMGEP repose sur les principes suivants : 1) auto-génération de buts comme fonctions de coût et sélection de buts basée sur des récompenses intrinsèques ; 2) exploration avec une recherche incrémentale de stratégies paramétrées par les buts et exploitation des données récoltées ; 3) réutilisation systématique de l'information obtenue pendant l'exploration de certains buts pour améliorer l'approche d'autres buts. Nous présentons en particulier une forme efficace de IMGEP qui utilise une représentation modulaire des espaces de buts ainsi que des récompenses intrinsèques basées sur le progrès en apprentissage. IMGEP est une architecture compacte et générale pour l'exploration de problèmes sans fonction objectif ou de problèmes où une telle fonction est difficile à définir et optimiser, alors que l'exploration motivée intrinsèquement permet une découverte efficace d'une diversité de solutions.

Nous évaluons l'architecture IMGEP dans plusieurs environnements de grande dimension avec utilisation d'outils. L'architecture IMGEP génère automatiquement un curriculum d'apprentissage efficace en données, ce que nous démontrons dans plusieurs environnements expérimentaux dont un avec un robot humanoïde qui explore de multiples espaces de buts avec des centaines de dimensions continues. Bien qu'aucun objectif en particulier ne soit fourni à ce système, le curriculum construit de façon autonome permet la découverte de compétences qui servent de tremplin pour l'apprentissage de compétences plus avancées, comme l'utilisation d'outils imbriquée. Nous démontrons que l'apprentissage de divers espaces de buts avec des motivations intrinsèques est plus efficace pour l'apprentissage de compétences complexes que de seulement s'intéresser à l'apprentissage de ces compétences. Certains aspects de cette architecture méritent d'être étudiés davantage, par exemple par une utilisation de modèles inverses plus sophistiqués, ou une comparaison avec les approches par recherche de nouveauté.

En résumé, nous avons mis en lumière l'impact des motivations intrinsèques dans certaines expériences de psychologie du développement et l'importance de bien les prendre en compte dans l'interprétation de ces expériences et dans les modèles du développement de l'enfant. Nous avons conçu un premier modèle robotique du développement intrinsèquement motivé de l'utilisation d'outils et avons montré que l'exploration orientée vers des buts avec des récompenses intrinsèques peut entraîner des trajectoires développementales qui ont des similarités avec le développement de l'enfant. Nous avons implémenté un premier modèle du développement de la parole à partir de zéro, dans un scénario de jeu naturel avec un parent, résultant en un apprentissage de la production de vocalisations qui ont un sens dans l'environnement et qui sont utilisées comme un outil social pour faire réagir et aider un parent. Nous avons aussi fourni un cadre algorithmique pour l'implémentation de processus intrinsèquement motivés d'exploration orientée vers des buts, qui est à la fois compact

et général. Ce formalisme a été implémenté et étudié de façon extensive dans différents environnements robotiques incluant un robot humanoïde. Nos agents robotiques ont développé un curriculum d'apprentissage de façon autonome grâce aux motivations intrinsèques dans l'exploration de l'utilisation d'outils tels que des joysticks qui contrôlent d'autres objets. Cependant, différents aspects du développement de l'enfant ne sont pas capturés par nos modèles. Cela inclut par exemple les contraintes maturationnelles sur le corps et la cognition, ou la complexité du guidage et de la contingence des parents. Nous pensons que ce travail représente un premier pas intéressant dans les directions de la compréhension des motivations intrinsèques chez les enfants et de leur implémentation pour améliorer l'apprentissage robotique de compétences avancées.

La modélisation de certains aspects du développement de l'enfant à travers la robotique permet de tester des hypothèses sur leurs mécanismes en expérimentant les comportements d'agents artificiels dotés de ces mécanismes. Cette modélisation peut permettre aux psychologues du développement d'affiner leurs hypothèses et de développer des tâches expérimentales plus adaptées, ce qui par la suite peut amener de nouvelles données qui peuvent servir à affiner les modèles robotiques. Poursuivre cette approche pourrait aider à répondre à ces questions toujours ouvertes : Comment les bébés et enfants sélectionnent leurs buts et stratégies ? Comment ce choix peut dépendre de leur expérience ? Comment des facteurs extérieurs comme le guidage social peuvent interagir avec ces motivations intrinsèques ?



Summary

Babies and children are curious, active explorers of their world. They play with the different objects and peers in their environment, without being asked to and perhaps until they fall asleep if we don't stop them. This intrinsic motivation to explore and learn new skills and facts seem present across all their learning situations and could be one of the important mechanisms of development. However, the underlying mechanisms of curiosity and intrinsic motivations have been little studied in developmental psychology, and intrinsic motivations are often neglected in the interpretation of child experiments. The mechanisms of intrinsic motivations such as how infants select their goals and strategies and how this interacts with external guidance are open questions.

Most artificial agents and robots have been learning in a way very different from humans. Some require that a human engineer specifies the objective of each particular task, others need extensive expert knowledge to guide learning, and many need millions of training samples. On one hand, modeling intrinsic motivations of children by implementing them in robotic agents could help us understand their mechanisms. On the other hand, designing artificial agents that learn and develop like humans, through the autonomous exploration and learning of diverse skills could improve the speed, robustness and adaptability of their learning in particular when human guidance is unavailable.

One of the challenges babies start to face early on is the learning of the relations between their actions and the changes in their environment. This learning takes place in an active manner, with the baby experimenting actions and observing the results as a little scientist. Those actions can be movement of parts of its body such as its limbs or its vocal tract, and the results can be changes in the sensory perception in any modality. Learning the relations between arm movements and the objects in the environment or between its vocal tract and the produced sounds and reactions of the peers involve experimenting and understanding the causality between all the different objects and sensory perceptions. The emergence of tool use and language in the first years of life is in great part a mystery, and little is known on the possible links between both skills. The concept of intrinsic motivations has not usually been considered in the development of those skills, although intrinsic motivations seem to be playing a key role. The development of tool use and of speech in infants and in robots is thus an interesting object of study for a better understanding of intrinsic motivations.

The objective of this thesis is twofold: understanding the role of intrinsic motivations in human development of speech and tool use through robotic modeling, and

improving the speech and tool-use learning abilities of artificial agents inspired by the mechanisms of human exploration and learning.

A first part of this work concerns the understanding and modeling of intrinsic motivations. Many experiments in developmental psychology evaluate particular skills of children by setting up a task that the child is encouraged to solve. However, children may sometimes be following their own motivation to explore the experimental setup or other things in the environment. We suggest that considering the intrinsic motivations of children in those experiments could help us understand their role in the learning of related skills and on long-term development. To illustrate this idea, we reanalyze and reinterpret a typical tool-use experiment aiming to evaluate particular skills in infants. We show that their motivations are diverse and do not always coincide with the target goal expected and made salient by the experimenter. Intrinsically motivated exploration seems to play an important role in the observed behaviors and to interfere with the measured success rates. However, our analysis has its own limits and it would be interesting to study intrinsically motivated behaviors in a dedicated setup triggering more diverse goals and strategies, with more experimental trials and with some automated data recording.

In order to model the development of tool use in the first years of life, we then define an intrinsically motivated artificial agent that generates its own goals and selects them based on intrinsic rewards, with a 2D simulated arm interacting with objects. With this model, we study how the particular implementations of intrinsic motivations to self-generate interesting goals together with the particular representation of goals can play a role in the tool-use progression. We show that an intrinsic motivation based on the learning progress to reach goals with a modular representation can self-organize phases of behaviors in the development of tool-use precursors that share properties with child development. This intrinsic motivation is compatible with observations in typical tool-use experiments with young children, but on the other hand intrinsic motivations are usually neglected in the interpretation of those experiments.

Several studies hypothesize a strong interdependence between speech and tool-use development in the first two years of life. To help us understand the underlying mechanisms, we present the first robotic model learning both speech and tool use from scratch. This model does not assume capabilities for complex action sequencing and combinatorial planning which are often considered necessary for tool use. Yet, we show that the learner progressively discovers how to grab objects with the hand, to use objects as tools to reach further objects, to produce vocal sounds, and to leverage these vocal sounds to use a caregiver as a social tool to retrieve objects. The discovery that certain sounds can be used as a social tool further guides vocal learning. This model predicts that the grounded exploration of objects in a social interaction scenario accelerates infant vocal learning of sounds to name these objects as a result of a goal-directed exploration of objects. However, those tool-use and speech learning models assume an already developed perception, while in the early development of those abilities the perception is continuously improving. Also, those

models would benefit from more direct and accurate comparisons with experimental data to refine their mechanisms.

In the second part of this thesis, we extend, formalize and evaluate the algorithms designed to model child development, with the aim to obtain an efficient learning agent that requires little expert knowledge and can adapt to new learning situations in an open-ended learning. We consider in particular goal babbling architectures that were designed to explore and learn solutions to fields of sensorimotor problems. However, so far these architectures have not been used in high-dimensional spaces of effects. We show the limits of existing goal babbling architectures for efficient exploration in such spaces, and introduce a novel exploration architecture called Model Babbling (MB). MB efficiently exploits a modular representation of the space of effects, and an active version of Model Babbling (AMB) further improves learning. These architectures are compared in a simulated experimental setup with an arm that can discover and learn how to move objects using several tools, embedding structured high-dimensional continuous motor and sensory spaces.

We then formalize an algorithmic approach called Intrinsically Motivated Goal Exploration Processes (IMGEP) that enables the discovery and acquisition of large repertoires of skills through self-generation, self-selection, self-ordering and self-experimentation of learning goals. The IMGEP algorithmic architecture relies on several principles: 1) generation of goals as fitness functions and selection of goals based on intrinsic rewards; 2) exploration with incremental goal-parameterized policy search and exploitation of the gathered data; 3) systematic reuse of information acquired when targeting a goal for improving towards other goals. We present a particularly efficient form of IMGEP that uses a modular representation of goal spaces as well as intrinsic rewards based on learning progress. IMGEP is a compact and general framework for the exploration of problems with no objective function or where an objective function is hard to define and optimize, while intrinsically motivated exploration allows an efficient discovery of a diversity of solutions.

We evaluate the IMGEP architecture in several high-dimensional tool-use environments. The IMGEP architecture automatically generates a sample-efficient learning curriculum within several experimental setups including one with a humanoid robot that can explore multiple spaces of goals with several hundred continuous dimensions. While no particular target goal is provided to the system, this curriculum allows the discovery of skills that act as stepping stones for learning more complex skills, e.g. nested tool use. We show that learning diverse spaces of goals with intrinsic motivations is more efficient for learning complex skills than only trying to learn these skills. Many aspects of this learning architecture are left for investigation in future work, such as the use of more accurate inverse models or a comparison with Novelty Search approaches.

To sum up, in this thesis we brought to light the impact of intrinsic motivations in child experiments and the importance of considering them in the interpretation of those experiments and in models of child development. We designed the first robotic models

of the intrinsically motivated development of tool use and showed that an exploration driven by goals and intrinsic rewards can result in developmental trajectories that have similarities with the development of infants. We implemented a first model of the development of speech from scratch in a naturalistic play scenario with a caregiver, resulting in the learning of the production of sounds that have a meaning in the environment and are used as a social tool to make the caregiver help. We also provided a formal algorithmic framework for the implementation of intrinsically motivated goal exploration processes that is compact and general. This framework has then been extensively studied and evaluated in several settings including a real robotic environment. Through intrinsic motivations, the robot autonomously developed a learning curriculum to explore a tool-use setup where joysticks can be used to act on other objects. However, many aspects of child development are not captured yet in our models. They include the maturational constraints on the body and cognition, or the complexity of the guidance and contingency of caregivers. Still, we believe this work provides interesting steps in the directions of the understanding of intrinsic motivations in children and of their implementation to improve robotic learning of advanced skills.

Modeling particular aspects of child development allows to test hypotheses about their mechanisms by experimenting the behavior of artificial agents endowed with those mechanisms. In turn, this modeling may help developmental psychologists to refine their hypotheses and their experimental setups, which then can bring new data to help us refine the robotic models. Following this approach could help us answer the remaining questions: how do babies select their goals and strategies? How does this choice depend on their previous experience? How do extrinsic factors such as caregiver's guidance interplay with intrinsic motivations?

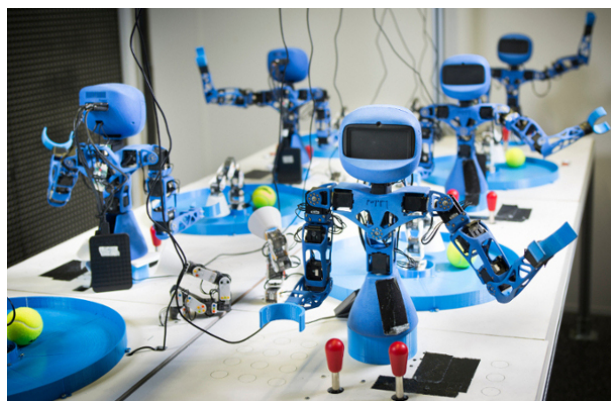


Table of Contents

1	Introduction	1
1.1	Intrinsic Motivations in Children	1
1.2	Artificial Intelligence and Robotics	3
1.3	Speech and Tool-Use Development	5
1.4	Objectives and Approach	6
1.5	Contributions and Outline	8
2	Background	13
2.1	Intrinsic Motivations and Curiosity	13
2.2	Tool-Use Development in Infants	20
2.3	Speech Development in Infants	27
2.4	Links between Tool-Use and Speech Development	31
2.5	Intrinsic Rewards and Motivations in Artificial Agents	34
2.6	Tool-Use and Speech Learning in Artificial Agents	44

Part I Intrinsic Motivations in Child Development 51

3	Intrinsic Motivations: Impact in Child Experiments and Role in Child Development	53
3.1	A Sequential Tool-Use Experiment with 21-Month Olds	56
3.2	Anecdotal Observations of Intrinsically Motivated Behaviors	61
3.3	Fine-grained Analysis of Behaviors and Events	66
3.4	Discussion	73
4	Modeling Intrinsic Motivations in the Development of Tool Use	81
4.1	Developmental Trajectories in Tool Use	84
4.2	Overlapping Waves of Strategy Preferences	94
4.3	General Discussion	109
5	A Unified Model of Speech and Tool-Use Early Development	113
5.1	Methods: a Grounded Play Scenario with a Caregiver	117
5.2	Results: Learning Tools and Words	122
5.3	Discussion	125

Part II	Intrinsically Motivated Artificial Intelligence	131
6	Modular Active Curiosity-Driven Discovery of Tool Use	133
6.1	Exploration Architectures	136
6.2	Tool-Use Simulated Environment	143
6.3	Exploitation Architectures	145
6.4	Results	145
6.5	Discussion	154
7	Intrinsically Motivated Goal Exploration Processes	157
7.1	Intrinsically Motivated Goal Exploration Processes	159
7.2	Modular Population-Based IMGEP	164
7.3	Relations to Related Work	167
7.4	Discussion	171
8	Experimental Study of IMGEP	175
8.1	Tool-Use Environments	176
8.2	Implementation of the IMGEP architecture	182
8.3	Results	184
8.4	Discussion	202
<hr/>		
9	Perspectives	207
9.1	Impact of Intrinsic Motivations on the Interpretation of Experiments	207
9.2	Experimental Paradigms for Studying Intrinsic Motivations	209
9.3	Tool Use and Language: an Evolutionary Developmental Perspective	211
9.4	IMGEP: Follow-ups and Relatives	216
9.5	IMGEP: Next Steps and Challenges	218
<hr/>		
Appendix A	Tool-Use Experiment: Ethograms	221
Appendix B	Learning a Representation for Goal Babbling	229
	Bibliography	253

Chapter 1

Introduction

Babies and children are curious, active explorers of their world. They play with the different objects and peers in their environment without being asked to and perhaps until they fall asleep if we don't stop them. This intrinsic motivation to explore and learn new skills and facts seem present across all their learning situations and could be one of the important mechanisms of development. However, the underlying mechanisms of intrinsic motivations have been little studied in psychology, and intrinsic motivations are often neglected in the interpretation of child experiments. The mechanisms of intrinsic motivations such as how infants select their goals and strategies and how this interacts with external guidance are open questions.

On the other hand, artificial agents and robots learn in a way very different from humans. Some require that a human engineer specifies the objective of each particular task, others need extensive expert knowledge to guide learning, and many need millions of training samples.

Modeling the intrinsic motivations of children by implementing them in robotic agents could benefit both the understanding of their mechanisms in children, and the design of artificial agents where the autonomous exploration of diverse skills could improve the speed, robustness and adaptability of their learning in particular when human guidance is unavailable. Those are the topics of this thesis.

1.1 Intrinsic Motivations in Children

Piaget studied the intelligence of children and documented its adaptation through the first years of life (Piaget, 1952), from the experimentation of simple actions in the environment to the invention of new means to achieve a goal. He considered that children possess inherent functions for cognitive adaptation, such as *assimilation*, whereby a new phenomenon is integrated in the learned schemas, and *accommodation*, modifying internal schemas to encompass the new situations. In Piaget's view, those functions operate from the inside and do not necessarily need to be triggered by a caregiver or other external events. This idea is for instance illustrated in the fifth stage of Piaget's classification of child development, which concerns the search and discovery of novel *means* through active experimentation. In one of the observations he

reported, his daughter Jacqueline experiments a new mean to retrieve an out-of-reach object, from 9-month to 12-month old, in 1926:

“Observation 149.-As early as 0;9 (3), Jacqueline discovers by chance the possibility of bringing a toy to herself by pulling the coverlet on which it is placed. She is seated on this coverlet and holds out her hand to grasp her celluloid duck. After several failures she grasps the coverlet abruptly, which shakes the duck; seeing that she immediately grasps the coverlet again and pulls it until she can attain the objective directly. [...]

Until 0;11 Jacqueline has not again revealed analogous behavior. At 0;11 (7), on the other hand, she is lying on her stomach on another coverlet and again tries to grasp her duck. In the course of the movements she makes to catch the object she accidentally moves the coverlet which shakes the duck. She immediately understands the connection and pulls the coverlet until she is able to grasp the duck.

During the weeks that follow Jacqueline frequently utilizes the schema thus acquired but too rapidly to enable me to analyze her behavior. At 1;0 (19) on the other hand, I seat her on a shawl and place a series of objects a meter away from her. Each time she tries to reach them directly and each time she subsequently grasps the shawl in order to draw the toy toward herself. The behavior pattern has consequently become systematic”

In Piaget’s observation, the baby is seen as an experimenter actively trying new actions and remembering little by little the ones that work best to reach particular goals. Piaget did not show the baby how to retrieve the toy or even attract the attention of the baby towards a possible solution. Neither did he train her by giving a reward when an attempt was successful. In other words, his child was intrinsically motivated, or curious to explore the objects in her environment, independently of any external incentive.

The concept of *intrinsic motivations*, also called *curiosity-driven learning*, caught some attention in psychology and has been studied both in animals and humans (Berlyne, 1960; Hunt, 1965; Kagan, 1972; Loewenstein, 1994; White, 1959). This line of research offered an alternative framework to Skinner’s behavioral theory (Skinner, 1953) for interpreting behavioral observations. Skinner indeed emphasized the role of extrinsic reinforcements and operant conditioning in learning. Psychologists first started to explore the definitions and properties or dimensionality of *curiosity* (see Loewenstein (1994) for a review). Berlyne categorized curiosity along two dimensions, distinguishing perceptual curiosity, a drive for novel stimuli, from epistemic curiosity, a desire for knowledge, and specific curiosity, a desire for a obtaining a particular target, from diversive curiosity, the general seeking of novelty (Berlyne, 1960). Those two dimensions of curiosity resulted in four different categories, e.g. the curiosity of a scientist searching for the solution of a well-defined problem is specific and epistemic, while Jacqueline’s may be specific and perceptual. Later, Ryan and Deci (2000)

formally defined intrinsic motivations as “the doing of an activity for its inherent satisfactions rather than for some separable consequence”.

Several mechanisms have been proposed to model intrinsic motivation triggered by a particular situation, activity or target, suggesting that the motivation is maximal when an intermediate level of incongruity, violation of expectations, or knowledge gap arise in that situation (see more details in the Background chapter). However, few studies have investigated those mechanisms in infants. Kidd et al. (2012) evaluated how the focus of attention in the environment of 8-month olds can be modulated by the statistical complexity of stimuli. They show that infants have a Goldilocks preference: a preference for sequences neither too simple nor too complex. This result could be compatible with many different underlying mechanisms, and concerns only one particular situation where the baby passively observes stimuli on a screen. Several experiments have shown that being curious about a piece of information improves the memory of this information once obtained (Gruber et al., 2014; Jepma et al., 2012; Kang et al., 2009). Begus et al. (2014) assess in babies how learning depends on their active role in the interaction with a caregiver. They present two novel objects to 16-month olds, then wait for the baby to point at one of the two objects. They show that learning is facilitated when the caregiver responds to pointing.

The active exploration of curious babies seems to be a fundamental mechanism of learning, however their exact mechanisms and the role of intrinsic motivations on the long-term development of children remain open questions. One particularity of the active exploration of infants that is observed as early as in newborns is that many of their actions seem goal-directed and not just reflexes or random actions (Von Hofsten, 2004). If infants are directing their actions towards goals, how do they select their current one? When do they switch goals? How do they choose their strategies or invent new ones? How does their previous experience with the learning situations affects their choice of goal and strategy? How do intrinsic motivations interplay with the external guidance of a parent?

1.2 Artificial Intelligence and Robotics

The chess-playing computer Deep Blue defeated the world champion Garry Kasparov in 1997, marking one of the well-known early successes of Artificial Intelligence. The AI of this computer involved a symbolic processing of the chess situations embedding rules and parameters defined by expert chess players together with computational power to simulate many moves in advance. However this algorithm was tuned to the task of playing chess such that to be able to play another type of game, the whole process of defining the rules and fine tuning the parameters would need to be done again by human experts and programmers.

If we imagine designing intelligent programs and robots that would potentially live in the home of users, then the programmers could not predict all the situations

that those systems will encounter and could not program all the associated skills. Oliver Selfridge pointed out that instead of expert systems filled with rules and parameters, “The essence of AI is learning and adapting” (Selfridge, 1993). By giving a machine the ability to learn the appropriate actions in new situations, that machine would get much closer to human intelligence. Many works in Machine Learning have focused on the design of neural network architectures and their learning algorithms. Recently, Deep Learning has offered impressive improvements in the performance of neural networks. In the domain of vision, the deep convolutional neural network of Krizhevsky et al. (2012) greatly improved the performances of image classification. Deep Reinforcement Learning algorithms enabled super-human performances in many Atari games (Mnih et al., 2013). The combination of deep neural networks trained with expert knowledge and reinforcement learning allowed to defeat professional players in the game of Go (Silver et al., 2016).

However, in all those examples, the artificial agents and robots learn in a way very different from humans. They require that a human engineer specifies the objective of each particular task to learn, or need extensive expert knowledge to bootstrap and guide learning (Silver et al., 2016). Also, many need millions of training samples if not more, or a huge database to learn patterns in a passive way (Krizhevsky et al., 2012).

Another approach to the building of machines surpassing human intelligence is to build *child* machines and to focus on the implementation of learning mechanisms inspired by the ones of human babies and children. One of the first occurrence of this idea is offered by Alan Turing after the description of its famous test of machine intelligence (Turing, 1950):

In the process of trying to imitate an adult human mind we are bound to think a good deal about the process which has brought it to the state that it is in. We may notice three components.

- (a) The initial state of the mind, say at birth,*
- (b) The education to which it has been subjected,*
- (c) Other experience, not to be described as education, to which it has been subjected.*

Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child’s? If this were then subjected to an appropriate course of education one would obtain the adult brain.

Implementing learning mechanisms in robots inspired by the ones of children is now part of the field of Developmental Robotics (Cangelosi et al., 2015; Lungarella et al., 2003). Many facets of development have been studied in robots with an inspiration from child research, such as sensorimotor coordination (Berthouze et al., 1996; Pfeifer and Scheier, 1997), self-exploration (Berthouze et al., 1998; Lungarella and Berthouze, 2003), social interaction (Breazeal and Scassellati, 1998; Dautenhahn

and Billard, 1999; Kozima and Yano, 2001; Kuniyoshi et al., 2003; Nagai et al., 2002). However, “simulating the child’s mind” to “obtain the adult brain” turned out to be presenting many challenges (Cangelosi et al., 2010), one of which being that the learning mechanisms of children are not completely elucidated.

1.3 Speech and Tool-Use Development

One of the challenges babies start to face early on is the learning of the relations between their actions and the changes in their environment. In their first years of life, they are able to develop complex skills by interacting with their environment and their peers. Speech and tool use are two hallmarks of intelligence, but their emergence is in great part a mystery, and little is known on the possible links between both skills.

A consistent observation in the development of the interaction with objects and the use of tools has been that babies actively explore, or *play*, and learn through this process. They can discover the use of tools accidentally, transfer their skills from one situation to another, or make use of demonstrations from social peers. When many objects can be explored in many different ways, goal-directed behaviors have been assumed to play a role in the selection of interesting actions by babies, once their cognitive abilities would allow them to retain a goal in memory (Guerin et al., 2013). Willatts (1990) argues that goal-directed behaviors in the play with external objects appear as early as 3 months where babies seem to be able to hold a complex goal in memory. However the mechanisms of the self-generation of goals have been little studied so far, particularly in the context of tool use. In many tool-use problem-solving studies, the fact that a baby would pursue a particular goal, such as retrieving an out-of-reach toy, and keep that goal throughout the study trial has been assumed and used to infer the tool-use capabilities. For instance, a very salient toy is placed out of reach and a non-salient tool is within reach in Rat-Fischer et al. (2012). However, babies could prefer other goals that look interesting to them because of their particular learning history, preferences for colors and shapes, for strategies, etc.

In parallel during their first year of life, babies also progressively learn to manage their vocal tract to go from producing squeals, growls or quasi-vowels to producing the vowels and speech-like syllables of their native language (Oller, 2000). Everyday, they produce spontaneously many sounds and vocalizations, even if we don’t trigger them. Around the age of 6-7 months, canonical babbling starts to appear in their vocalizations. Canonical babbling is the production and repetition of syllables with one consonant and one vowel that are the building blocks of words in languages. It precedes the production of the first spoken words, and is considered critical for the learning of speech. Babies progressively learn to produce the sounds of the particular language of their environment through mechanisms such as the imitation of ambient sounds or the interaction with social peers. However, the role of intrinsic motivations

in spontaneous vocal exploration, and the interaction between this autonomous exploration and social interaction have been little studied (Moulin-Frier et al., 2013).

Tool use and speech seem to require similar information processing capabilities allowing the production and perception of sequential combinations of increasing complexity, from reaching to spoon self-feeding and from phonemes to stories. Several parallels have been drawn between these two abilities in their ontogenetic development (Greenfield, 1991; Meltzoff, 1988), and they could use similar neural substrates (Higuchi et al., 2009). Tool use and speech could also have a related evolutionary origin, with several proposed scenarios (Greenfield, 1991; Morgan et al., 2015).

Speech and tool use are thus two fundamental skills that start to develop in the first years of life, through a combination of several learning mechanisms. Intrinsically motivated exploration seems to be one of their key learning mechanisms, but have received little attention so far.

1.4 Objectives and Approach

The question of the role and functioning of intrinsic motivation is a fundamental one for the understanding of long-term child development. Intrinsic motivations are driving exploration and learning but their particular mechanisms are not understood. How do children generate and select goals, and strategies to reach their goals? Do the mechanisms of curiosity evolve across development? Do they depend on the learning situations, on the action or sensory modalities? Two particular domains, speech and tool-use development, largely involve intrinsic motivations, but their particular role in the development of those skills need investigation. Furthermore, most artificial agents and robots learn with given objectives, require extensive expert knowledge, or millions of iterations, such that their learning seems far from human's. Would intrinsic motivations improve their learning in some respects?

On one hand, modeling intrinsic motivations of children by implementing them in robotic agents could help us understand their mechanisms, enabling the embodied experimentation and evaluation of different hypotheses for those mechanisms. On the other hand, designing artificial agents that learn and develop like humans, through the autonomous exploration and learning of diverse skills could improve the speed, robustness and adaptability of their learning in particular when human guidance is unavailable. Recent work in the field of developmental robotics started to implement the concepts of intrinsic motivations (Oudeyer et al., 2007) and of the robotic exploration with self-generated goals (Baranes and Oudeyer, 2010a; Rolf et al., 2010) with an inspiration from the intrinsic motivations in child development. One family of models has considered a curiosity-driven learning mechanism where the learner actively engages in sensorimotor activities that provide high learning progress, avoiding situations that are too easy or too difficult and progressively focusing on activities of increasing complexity (Gottlieb et al., 2013). Such computational models have shown

that developmental trajectories could emerge from the curiosity-driven learning of sensorimotor mappings, in several different settings. In the Playground Experiment (Oudeyer et al., 2007), a quadruped robot motivated to maximize its learning progress discovered the use of its motor primitives to interact with the items of an infant play mat and a robot peer, and followed a self-organized learning curriculum. In a model of active vocal development (Moulin-Frier et al., 2013), an agent learned how to produce sounds with its vocal tract by self-exploration combined with imitation of adult speech sounds. This model reproduces major phases of infant vocal development until 6 months. In both studies, developmental trajectories are emerging from learning, with both regularities in developmental steps and diversity in the development of several independent learners. However, in previous work implementing intrinsic motivations and goal-directed exploration in robots, a single space of goal is explored, either discrete or low-dimensional. In order to model the development of tool use and speech in realistic scenarios, more sophisticated representations of goals and exploration algorithms are required.

The objective of this thesis is twofold: understanding the role of intrinsic motivations in human development of speech and tool use through robotic modeling and experimentation, and improving speech and tool-use learning abilities of robots inspired by the mechanisms of human exploration and learning.

Modeling particular aspects of child development through robotic models allows to test hypotheses about their mechanisms by experimenting the behavior of artificial agents endowed with those mechanisms. In turn, this modeling may help developmental psychologists to refine their hypotheses and their experimental setups, which then can bring new data to help us refine the robotic models. Following this approach could help us answer questions such as: how do babies select their goals and strategies? How does this choice depend on their previous experience? How do extrinsic factors such as caregiver’s guidance interplay with intrinsic motivations?

In this thesis, we study in more details the impact of intrinsic motivations in tool-use experiments in developmental psychology. We extend previous models of intrinsically motivated goal exploration (Baranes and Oudeyer, 2010a; Moulin-Frier et al., 2013) to allow the learning of complex skills such as tool use and speech in high-dimensional settings that model more closely the natural environments of children. We design naturalistic environments where the fact that some objects can be used as tools is not assumed and has to be discovered, and we study a modular representation with many spaces of high-dimensional continuous goals that are related to the objects of the environment. The learning of speech is grounded in an environment where the produced sounds have a meaning related the objects, in a language spoken by peers. We compare several variants of implementations of intrinsic motivations to understand their impact on learning and development of robots and to see which implementations are compatible with the observed behaviors of children. We then study the efficiency of learning through intrinsically motivated goal exploration in diverse tool-use robotic environments including physical robots.

1.5 Contributions and Outline

A first part of this work concerns the understanding and modeling of intrinsic motivations. Many experiments in developmental psychology evaluate particular skills of children by setting up a task that the child is encouraged to solve. However, children may sometimes be following their own motivation to explore the experimental setup or other things in the environment. We suggest that considering the intrinsic motivations of children in those experiments could help understanding their role in the learning of related skills and on long-term child development. To illustrate this idea, in chapter 3 we reanalyze and reinterpret a typical tool-use experiment aiming to evaluate particular skills in infants. We show that their motivations are diverse and do not always coincide with the target goal expected and made salient by the experimenter. Intrinsically motivated exploration seems to play an important role in the observed behaviors and to interfere with the measured success rates. However, intrinsic motivations are usually neglected in the interpretation of this kind of experiments.

In order to model the development of tool use in the first years of life, we then define in chapter 4 an intrinsically motivated artificial agent that generates its own goals and selects them based on intrinsic rewards, with a 2D simulated arm interacting with objects. With this model, we study how the particular implementations of intrinsic motivations to self-generate interesting goals together with the particular representation of goals can play a role in the tool-use progression. We show that an intrinsic motivation based on the learning progress to reach goals with a modular representation can self-organize phases of behaviors in the development of tool-use precursors that share properties with child development.

Several studies hypothesize a strong interdependence between speech and tool use development in the first two years of life. To help us understand the underlying mechanisms, we present in chapter 5 the first robotic model learning both speech and tool use from scratch. This model does not assume capabilities for complex action sequencing and combinatorial planning which are often considered necessary for tool use. Yet, the learner progressively discovers how to grab objects with the hand, to use objects as tools to reach further objects, to produce vocal sounds, and to leverage these vocal sounds to use a caregiver as a social tool to retrieve objects. The discovery that certain sounds can be used as a social tool further guides vocal learning. This model predicts that the grounded exploration of objects in a social interaction scenario should accelerate infant vocal learning of accurate sounds for these objects' names as a result of a goal-directed exploration of objects.

In the second part of this thesis, we extend, formalize and evaluate the algorithms designed to model child development, with the aim to obtain an efficient learning agent that require little expert knowledge and can adapt to new learning situations in an open-ended learning. We consider in particular goal babbling architectures



Figure 1.1: The sequential tool-use task we analyzed. A salient toy is placed inside a transparent tube open on both ends. Wooden blocks are placed around the tube, at least two of which must be inserted on one side of the tube to push the toy out.

that were designed to explore and learn solutions to fields of sensorimotor problems. However, so far these architectures have not been used in high-dimensional spaces of effects. In chapter 6, we show the limits of existing goal babbling architectures for efficient exploration in such spaces, and introduce a novel exploration architecture called Model Babbling (MB). MB exploits efficiently a modular representation of the space of effects, and an active version of Model Babbling (MACOB) further improves learning. These architectures are compared in a simulated experimental setup with an arm that can discover and learn how to move objects using several tools, embedding structured high-dimensional continuous motor and sensory spaces.

We then formalize in chapter 7 an algorithmic approach called Intrinsically Motivated Goal Exploration Processes (IMGEP) that enables the discovery and acquisition of large repertoires of skills through self-generation, self-selection, self-ordering and self-experimentation of learning goals. The IMGEP algorithmic architecture relies on several principles: 1) self-generation of goals as fitness functions and selection of goals based on intrinsic rewards; 2) exploration with incremental goal-parameterized policy search and exploitation of the gathered data; 3) systematic reuse of information acquired when targeting a goal for improving towards other goals. We present a

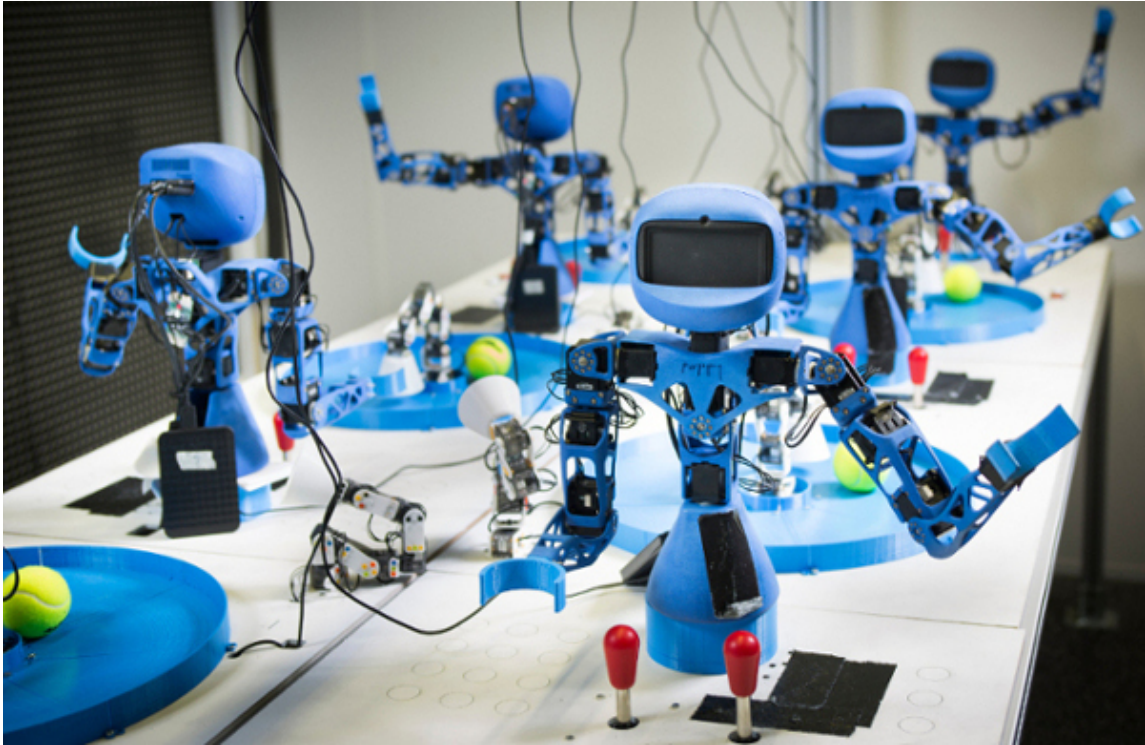


Figure 1.2: The robotic tool-use environment built in our experiments. A Poppy Torso robot (the learner) is mounted in front of two joysticks that can be used as tools to act on other objects: a Poppy Ergo robotic toy and a ball that can produce lights and sounds. Six copies of the same setup are running in parallel to gather more data.

particularly efficient form of IMGEP that uses a modular representation of goal spaces and a population-based policy. IMGEP is a compact and general framework for the exploration of problems with no objective function or where an objective function is hard to define and optimize, while the intrinsically motivated exploration allows an efficient discovery of a diversity of solutions.

In chapter 8, we evaluate the modular population-based IMGEP architecture in several high-dimensional tool-use environments. The IMGEP architecture automatically generates a sample-efficient learning curriculum within several experimental setups including a real humanoid robot that can explore multiple spaces of goals with several hundred continuous dimensions. While no particular target goal is provided to the system, this curriculum allows the discovery of skills that act as stepping stones for learning more complex skills, e.g. nested tool use. We show that learning diverse spaces of goals with intrinsic motivations is more efficient for learning complex skills than only trying to directly learn these complex skills.

To sum up, in this thesis we bring to light the impact of intrinsic motivations in child experiments and the importance of considering them in the interpretation of those experiments and in models of child development. We design the first robotic models of the intrinsically motivated development of tool-use and show that an exploration driven by goals and intrinsic rewards can result in developmental trajectories that have similarities with the development of infants. We implement a first model of the development of speech from scratch in a naturalistic play scenario with a caregiver, resulting in the learning of the production of sounds that have a meaning in the environment and are used as a social tool to make the caregiver help in different ways. We also provide a formal algorithmic framework for the implementation of intrinsically motivated goal exploration processes that is compact and general. This framework is then extensively studied and evaluated in several settings including a real robotic environment. Through intrinsic motivations, the robot autonomously develop a learning curriculum to explore a tool-use setup where joysticks can be used to act on other objects. We discuss the perspective of integrating and combining our modular goal exploration implementation with Deep Reinforcement Learning algorithms. However, many aspects of child development are not captured yet in our models. They include the maturational constraints on the body and cognition, or the complexity of the guidance and contingency of caregivers. Still, we believe this work provides interesting steps in the directions of the understanding of intrinsic motivations in children and of their implementation to improve robotic learning of advanced skills.

Chapter 2

Background

2.1 Intrinsic Motivations and Curiosity

2.1.1 Mechanisms of Intrinsic Motivations

Intrinsic motivations are a drive to explore, play, act, ask, combine, or build with no apparent reward, which in the process make learning happen. Once intrinsic motivations were observed, psychologists wondered what could be the mechanisms pushing humans and animals to be curious, and if they could uncover some of their properties through experimentations. One fundamental question is what triggers curiosity from an operational point of view: why some people prefer to explore a particular activity, say playing darts, to exploring an atlas? Why one person likes this activity at some point and not anymore later on? In which situation does a child prefer drawing versus playing video games and how does this preference depend on the particular social context and environment?

Many theories have been proposed to account for the mechanisms underlying intrinsic motivations. Piaget saw in children a need to make sense of the world. Curiosity in children would result from a discrepancy between their expectations and the reality, so that they prefer situations allowing them to assimilate new information into their learned schemas and accommodate their schemas to account for new experiments (Piaget, 1952). Following Piaget, the motivation of children would be more intense for an optimal level of discrepancy, whereas a too low discrepancy would make assimilation too easy, and a too high level of discrepancy would make them unable to relate the new situation to the known schemas (McCall and McGhee, 1977). Hebb conceptualized a preference for an optimal level of incongruity (Hebb, 1955), where a mismatch between expectations and perceptions is “pleasurable”, however a too incongruous situation is unpleasant. Hunt also postulated a search for intermediate levels of incongruity (Hunt, 1965), and Kagan extended his theory to include other motivations coming from a cognitive dissonance (Kagan, 1972): incompatibilities between ideas or between ideas and behaviors. In those views, the relation between the motivation and the incongruity, violation of expectations or dissonance has an inverted U shape, with an intermediate level giving rise to an optimal level of motivation, and low or high levels resulting in a lower motivation.

Loewenstein proposes a more general knowledge gap theory (Loewenstein, 1994), where curiosity is triggered by the identification of gaps of information in knowledge. Those gaps can be discovered through the occurrence of violated expectations, and feeling competent at a task can come from to the filling of those gaps.

There have been many experiments studying different aspects of curiosity in humans, starting with questionnaires measuring the curiosity of participants in diverse situations. Several measures of curiosity have been developed with questionnaires such as the Ontario Test of Intrinsic Motivation (Day et al., 1971), the Melbourne Curiosity Inventory (Naylor, 1981) or others (Kashdan et al., 2018; Langevin, 1971; Penney and McCann, 1964). One topic of interest has been the question of whether curiosity is correlated with measures of intelligence and creativity. The results were not always conclusive as some studies found a small correlation and others did not (Langevin, 1971; Penney and McCann, 1964; Voss and Keller, 1983), see (Loewenstein, 1994) for a review. However, those results may have reached the limits of self-reported questionnaires, which through a list of indirect questions try to evaluate the curiosity of one person. First, the subjects may have wrong estimations of their own curiosity and related behaviors. They can also have different interpretation or scale for the meaning of *curious* in sentences like “Being curious about my classwork is important to me” in the Experimental Curiosity Measure of (Langevin, 1971). Furthermore, they may well understand that their curiosity is being assessed which can bias their answers. Those limits make the interpretation of the results of questionnaires even more problematic with children.

In order to study intrinsic motivations in babies and children, a more recent approach has been to observe their focus of attention and their exploratory actions in situations where multiple stimuli are available to them, depending on several factors hypothesized to be linked to curiosity, such as the novelty of stimuli, the affordances of available objects, the consistency of events, etc.

For instance, when children are shown evidence that supports an hypothesis which is considered as improbable by the infant, their curiosity can be triggered. Indeed, Bonawitz et al. (2012) show that 4 to 7 years-old children play more with an object when it is not behaving (falling) as expected, and are more likely to find a third object (a magnet) and to invoke it to explain the unexpected behavior. Those observations are thus compatible with hypotheses for curiosity mechanisms such as incongruity, violations of expectations or knowledge gap. In another experiment, they show that 4 to 5 years-old children explore more a box with levers that make puppets pop out of the box when they had confounded evidence about which levers action which puppet than when they had unconfounded evidence (Schulz and Bonawitz, 2007). Manipulating the degree to which candidate causes could be isolated and the potential for information gain of exploration, they show that children first understand how distinguishable the evidence is, and also are able to design interventions that generate distinctive patterns of evidence and maximize information gain (Cook et al., 2011).

By measuring the looking patterns of 8-month-old infants, Kidd and collaborators

(Kidd et al., 2012) evaluated how their focus of attention in the environment is modulated by the statistical complexity of stimuli. They presented 3 objects popping out of their respective boxes, in a sequence of varying complexity. The complexity of a sequence was measured by its information content: the negative log probability of an object popping out of a given box. Infant's attention to an object is measured by the time after which they look away from the object, given by eye-tracking tools. They show that infants have a Goldilocks preference: a preference for sequences neither too simple nor too complex. This Goldilock effect is compatible with the hypothesized inverted U shape between the motivation and the intensity of the incongruity, violation of expectations or cognitive dissonance (Loewenstein, 1994).

Those behavioral experiments give insights into particular aspects of curiosity in particular experimental contexts. They help delineate the contexts and situations where a particular curiosity mechanism could be at play in the observed behaviors. However, they form a relatively small body of results on the psychological underpinnings of curiosity in infants. Indeed, it is hard to isolate the curiosity component from other mechanisms in place such as memory, learning, or social incentives. Also, particular results are compatible with several general hypotheses which may not be detailed enough to be discriminated.

Much more recently, the mechanisms of curiosity and information seeking started to be studied from the neuroscientific point of view. Several neuroimaging techniques have been used to study the role of particular brain regions and particular neural cells in information seeking tasks. For instance, Kang et al. (2009) asked adults to report their curiosity level towards the answer of given questions and recorded the associated brain activations through functional magnetic resonance imaging (fMRI). They observed a correlation between the curiosity of subjects and the neural activation in both the lateral prefrontal cortex and the caudate. Based on previous findings on the role of those regions, the authors suggest that curiosity is an anticipation of rewarding information. In the context of perceptual curiosity, showing a blurred image triggers curiosity about the content of the original image. In a fMRI setup, Jepma et al. (2012) shows that triggering perceptual curiosity activated regions sensitive to conflict and arousal, while the relief of perceptual curiosity activated regions related to reward processing. The authors interpretation is that curiosity is an aversive state whose termination is rewarding.

In order to uncover the particular neural circuitry involved in curiosity and information seeking, a promising approach is the recording of the activation of single neural cells in monkeys during the execution of a visual task. Indeed, in foveate animals, eye movements can tell where they focus their attention or at least where they search for visual information. Studying single cells activations in visual areas of the monkey brain while recording eye saccades in an information seeking visual task can help to understand the precise functions of those individual cells (Gottlieb et al., 2013). In such a setup, Gottlieb and Balan (2010) shows that some neurons in the lateral intraparietal area (LIP) in monkeys encode the expected reward of

a saccadic eye movement, while other encode the expected information gain of the saccade. Those cells could be part of the neuronal circuitry involved in selecting eye movement actions that best fill information gaps triggered in the task.

2.1.2 Goal-Directed Behaviors

When a 6-month-old baby tries to reach for an out-of-reach toy, fails, cries, and finally get it through the help of its mother, one may assume that the baby was somehow having a goal in mind: getting the toy for mouthing or throwing it, and in the end succeeded to achieve this goal through trying several strategies. Goal-directed actions and behaviors seem to be constantly encountered in children free play (Von Hofsten, 2004). Actions have been defined by Dickinson and Balleine (1994) as goal-directed when the “performance is mediated by knowledge of the contingency between the action and the goal or outcome, whether this knowledge is conceived as an expectation or belief or an associative connection”. Goals have been widely discussed from a theoretical standpoint, elaborating on the possible structure of goals and the processes involved in goal representation and generation (Austin and Vancouver, 1996). One recurring proposition is that goals follow a hierarchical organization, with higher-level goals extended over a longer period of time than lower-level subgoals necessary to achieve the higher-level goals. However, little is known on the mechanisms of goal-directed behaviors in children and particularly on the neural substrates of the representation and sampling of goals in the brain. Here, we review neuroscientific evidence of the neural correlates of some aspects of goal-directed behaviors.

An interesting property of the neural correlates of goals in the brain is the involvement of the mirror neuron system (MNS) in their encoding (Fogassi et al., 2005). A mirror neuron is a neuron that activates both if a particular action is observed and if that action is executed. Recently, some mirror neurons have been shown, in monkeys, to be specific of the goal of the observed or produced action, instead of the action itself (Fogassi et al., 2005). In humans also, parts of the mirror neuron system, the inferior fronto parietal cortex, has been argued to represent the outcome of actions (Hamilton and Grafton, 2007). Human understanding of actions has been described as a hierarchical information processing system, with a cascade of specialized processes from occipital (sensory) to parietal (MNS) and frontal regions (Hamilton and Grafton, 2007; Thill et al., 2011), where the self and other actions in the world are represented both in terms of their consequences and of the intentions underlying them.

Farther in the information processing hierarchy, the medial prefrontal cortex also contributes to the representation of goals and subgoals, according to fMRI experiments on human adults (Fernandes et al., 2018). Participants had to move a truck displayed on a screen through the use of a joystick. By manipulating the distance of the goal and subgoals of the task, the authors show that the medial prefrontal cortex signals prediction errors related to subgoals independently of goals in some context, and

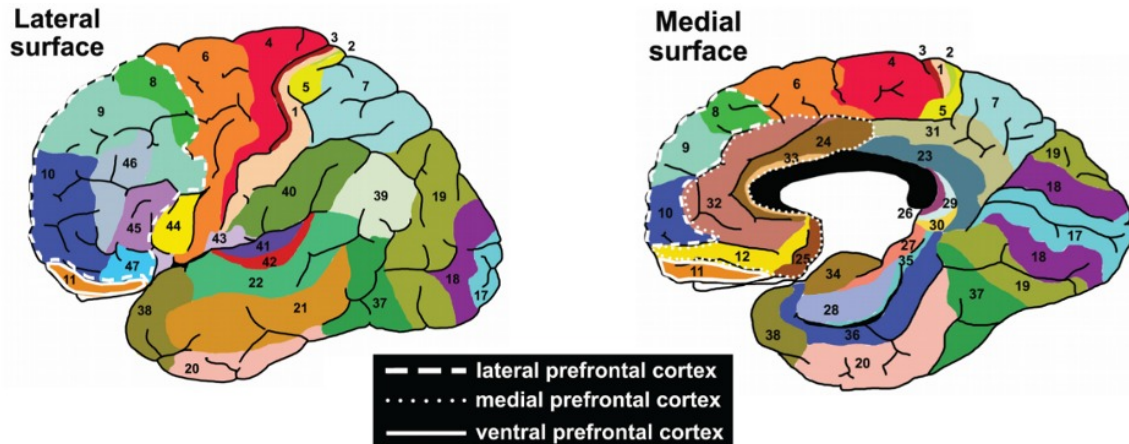


Figure 2.1: Map of the adult human cortex as categorized by Brodmann. Adapted from Wikimedia Commons.

signals prediction errors related to goals in other situations. They argue that the medial prefrontal cortex selectively attends to information at different levels of the goal and prediction errors hierarchies depending on the task context.

Those psychological and neuroscientific experiments aim to shed light on the real-time functioning of curiosity mechanisms in humans and animals. Another question of interest is the extent to which curiosity and information-seeking behaviors benefit learning.

2.1.3 Intrinsic Motivations and Learning

Several experiments have shown that being curious about a piece of information improves the memory of this information once obtained. In Kang et al. (2009), a high self-reported curiosity towards the answer of given questions led to a better recall of those answers. In particular, an unexpected answer (incorrectly guessed) when the curiosity was high resulted in an increased activation of memory areas and a better recall a few weeks later.

Those findings were replicated by Gruber et al. (2014) which also exhibit an improved memory for information subjects are curious about, with an involvement of the midbrain, the hippocampus and their interaction. They also show that high curiosity states improve incidental memory, the memory of independent objects shown before the relief of curiosity. They interpret their results as supporting a positive influence of being in a curious state on memory of new information. Furthermore, they hypothesize that intrinsic and extrinsic motivations and their influence on memory formation could share common neural mechanisms. In the context of perceptual curiosity, a high curiosity state triggered by a blurred image is also shown to increase incidental memory, through hippocampal activations (Jepma et al., 2012).

In the previous experiments, adults in fMRI scanners are passively learning what is shown to them. However, intrinsic motivations or curiosity are fully realized when many alternative behaviors are available to choose from. A study by Begus et al. (2014) aims to assess how learning of babies depends on their active role in the interaction with a caregiver. They present two novel objects to 16-month olds, then wait for the baby to point at one of the two objects, and in the first condition (congruent) they show what this object's function is, and in the other condition (incongruent) they show what the other object's function is. After a break, babies were given each object individually and were prompted to perform the action demonstrated before by the experimenter. The results show that babies replicated correctly significantly more the function of the object if they had pointed towards it (40.6% vs 12.5%). A follow-up experiment shows that when no choice was given to babies, their average correctly replicated actions is 12.2%. Together these experiments show that babies learning is facilitated when pointing is responded to. The active role of curious babies is thus a fundamental mechanism of the learning process.

2.1.4 Intrinsic Motivations and Development: An Evolutionary Perspective

From babies to adults, we have seen that exploring and being curious about a fact or a situation can benefit learning and memory thereof. However, little is known about the long-term consequences on ontogenetic development of being intrinsically motivated to explore and to learn. Do stronger intrinsic motivations and curiosity in childhood increase the cognitive abilities later on in development? Or do they make children waste time and energy in what turns out to be useless information-seeking behaviors on the long run? How does the interplay between intrinsic motivations and social guidance influence children developmental trajectories?

Almost no research studied these questions, one exception being a longitudinal study of baby gaze and pointing behaviors. Brooks and Meltzoff (2008) recorded in lab sessions the amount of following of parent's gaze, the amount of spontaneous non-elicited pointing and the vocabulary growth of babies from 10-month old to 2-years old. They find a positive relationship between the amount of gaze following and pointing at 10 month and the vocabulary growth during the second year. This result provides evidence, in a social interaction context, to the hypothesis that intrinsically motivated exploration has a positive influence on the long-term development of babies.

It has been hypothesized that intrinsic motivations may be serving as a filter on what to explore and learn in a complex environment with much more things to learn than possible in a lifetime (Gottlieb et al., 2013). Indeed, babies must cope with the vast complexity of the physical world they are born in, through the altricial and evolving body they are born with. This world contains caregivers and other peers making unknown sounds of an unknown language, together with many objects and tools of infinite functions. Actively exploring and learning as most skills and

knowledge as possible given time and environmental constraints during childhood might overall be a good strategy for development, furthermore when the skills needed to survive to and live in adulthood are initially unknown: some of the learned skills and knowledge may turn out later to be useful.

With an evolutionary perspective, several fundamental questions remain: what is the evolutionary origin of intrinsic motivations? How do intrinsic motivations increase the fitness of a species? Did intrinsic motivations influence the evolution of other cognitive abilities in humans?

Indeed, as curiosity and intrinsic motivations seem to benefit the long-term development of babies and children, we may hypothesize that they increase the evolutionary fitness of the human species. One particularity of the human species and to a lesser extent of other great apes, is the long period of protected development from childhood to adulthood (Power, 1999). The human species may have evolved in the direction of being less capable as a newborn, but more capable to learn during development, assisted by an increase in brain size and with a crucial role played by intrinsic motivations. In a computational model of learning, Singh et al. (2010) studied the role of intrinsic motivations within an evolutionary settings and show results that support this hypothesis. They formulated an evolutionary context where the environment changes over time, such that reinforcement learning agents maximizing an optimal reward function for a particular environment may not learn efficiently in other environments. They showed that a reward function reinforcing behaviors that are “ubiquitously useful across many different environments” can lead to a better evolutionary fitness than a function rewarding only behaviors targeted at survival and reproduction. The authors also argue that the difference between intrinsic and extrinsic rewards could be one of degree, where extrinsic motivations could be rewarding events related to the immediate survival and transmission of genes whereas intrinsic motivations could increase the evolutionary fitness on the very long term (Barto, 2013).

Furthermore, intrinsic motivations could have influenced the development and evolution of other cognitive abilities. In Oudeyer and Smith (2016), the authors explain that previous computational and robotic models of vocalization learning have shown that conventional patterns of vocalizations at the group level could emerge from the interaction of intrinsically motivated individuals. They argue that the evolution of language prerequisites and potentially the evolution of other cognitive abilities could have been facilitated by intrinsic motivations.

Passingham and Wise (2012) studied the evolution of the prefrontal cortex from early primates to anthropoids, and reconstructed the probable ecological niches of the human lineage. Through a comparative ecological approach together with the interpretation of recent neural data, in particular the interconnections of the different brain regions, they propose that “the granular PF cortex generates goals that are appropriate to the current context and current needs, and it can do so based on a single event”. They argue that from its connections, the granular prefrontal cortex can

represent three hierarchies: the context, goal, and outcome hierarchies. For instance, goals can be represented in a range of hierarchical levels, with goals such as the specification of an object or location used as a target of action, the specification of the abstract structure of a series of actions, or the specification of a rule or strategy that generates objects or locations to choose or to avoid. The prefrontal cortex has the ability to choose actions based on outcomes (medial PFC), choose objects based on outcomes (orbital PFC), search for goals (caudal PFC), generate goals based on recent events (dorsal PFC) and generate goals based on visual and auditory contexts (ventral PFC), such that as a whole, the PFC can generate goals from current contexts and events (see Fig. 2.1 for a map of the prefrontal cortex). In the successive ecological niches of primates, the PFC could have been used to link the foraging actions with the resources outcome that follow, link foraging goals (objects, places) with resource outcomes, select foraging targets, keep goals in memory, allow a fast learning to reduce wasteful and risky choices, and do mental trial and error. In the hominid lineage in particular, it could have supported teaching and learning by instruction with less errors, the imagination of more complex goals, the monitoring of others intentions, and improved reasoning abilities (Passingham and Wise, 2012).

Those results and hypothesis pave the way for more research to understand the precise mechanisms and functions of intrinsic motivations and their influence on human ontogenetic and phylogenetic development.

2.2 Tool-Use Development in Infants

2.2.1 Tool-Use Definitions

Many definitions of tool-use behaviors have been offered from behavior researchers. Most of them agree on the general idea of a tool being an object used to interact with another object. However, they all differ when delineating the limits of tool-use behaviors on several dimensions such as the origin the tool, the relation between the tool and the environment, the relation between the tool and the object acted upon, the efficiency of the behavior.

The early definition of Van Lawick-Goodall (1971) states that “a tool-using performance in an animal or bird is specified as the use of an external object as a functional extension of mouth or beak, hand or claw, in the attainment of an immediate goal. This goal may be related to the obtaining of food, care of the body, or repulsion of a predator, intruder, etc. If the object is used successfully, then the animal achieves a goal which, in a number of instances, would not have been possible without the aid of the tool.” This definition is one of the first to explicitly include the potential behaviors of some non-human animals.

Alcock explicitly only includes the use of inanimate objects as tools (Alcock, 1972): “Tool-using involves the manipulation of an inanimate object, not internally

manufactured, with the effect of improving the animal's efficiency in altering the form or position of some separate object." Those two definitions also emphasize that a tool must improve the efficiency of some behaviors compared to behaviors without that tool.

A definition commonly referred to is the one of Beck (1980): "the external employment of an unattached environmental object to alter more efficiently the form, position, or condition of another object, another organism, or the user itself when the user holds or carries the tool during or just prior to use and is responsible for the proper and effective orientation of the tool."

In this thesis, following Goodall and Alcock's definitions and contrary to Beck's, we will also consider as tools the manipulation of objects attached to the environment, such as a joystick that controls another object (see chapter 8). Beck also included "social tool use" in tool-use behaviors, contrary to Alcock's, as the manipulation of another individual as a tool (Bentley-Condit et al., 2010). In chapter 5, we discuss the emergence of social tool use in our experiments with a simulated agent producing vocalizations to make a caregiver move a toy.

When multiple tools interact, Wimpenny et al. (2009) suggest to use the term "meta-tool use". In this category, we can find for instance the use of a tool to retrieve another tool, which is called a sequential tool use, and the use of a tool to build another tool, called a constructive tool use.

2.2.2 Tool-Use Development

Tool-use behaviors in babies come after a long period of development of behaviors of increasing complexity with the hands and with objects, from the very first behaviors such as rooting for the breast, to the manipulation of sticks and spoons. The precursors of tool-use behaviors can be grouped into three categories (Guerin et al., 2013).

The first category concerns behaviors without objects. They include for instance many rhythmical stereotypical behaviors such as those observed by Thelen (1979) in a longitudinal study in the first year of life: arm waving, flexion and extension of fingers, rotation and flexion of the hand, clapping hands together, etc.

The second category includes behaviors with a single object. Reaching for and grasping an object are well studied behaviors with a single object. Many types of grasping behaviors are developed little by little in the first year, from the simplest palmar grasp to the more complex pincer grasps (grasping the object between two fingers) (Guerin et al., 2013). Grasping can be followed by the visual inspection of the object, or mouthing, throwing the object away, etc.

The third category deals with object-object behaviors, where several objects interact. It includes pushing or banging an object on a surface (table or floor), pulling a towel supporting another object at about 8 months (Willatts, 1999), fitting shapes into slots, acquired around 12 months (Örnkloo and von Hofsten, 2007), exploring the relations among objects, which is preferred at 13 $\frac{1}{2}$ months versus exploring objects

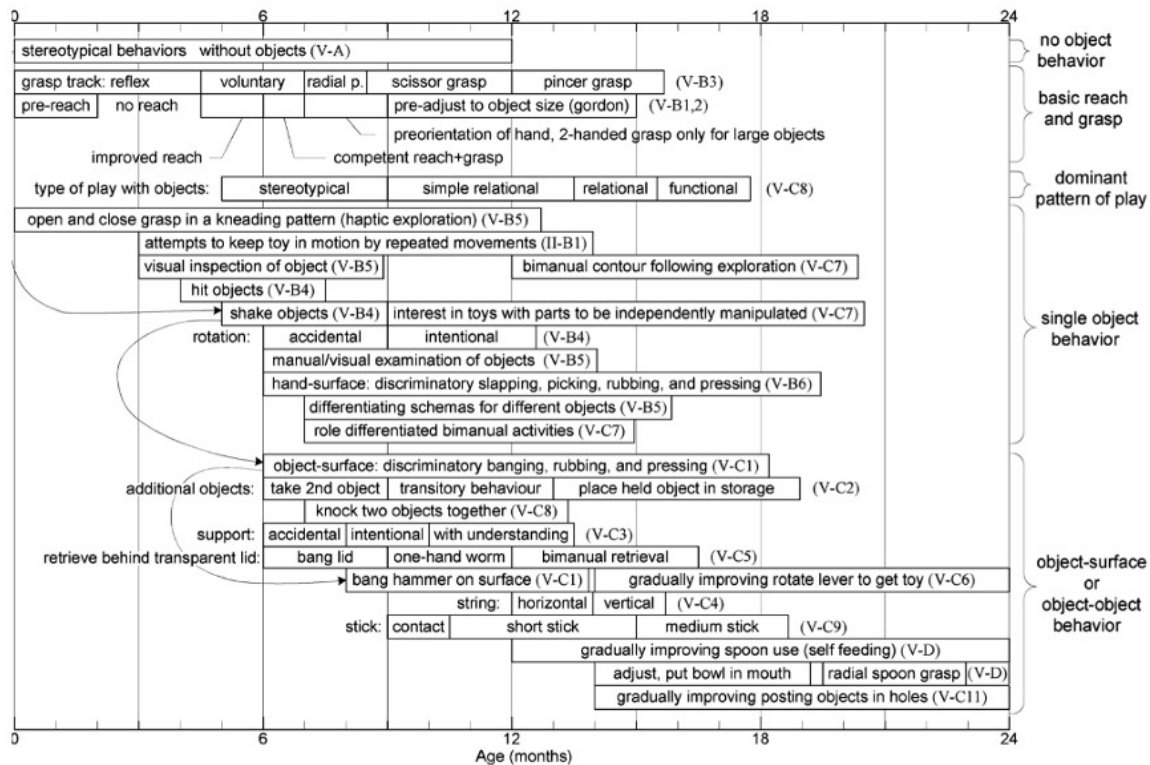


Figure 2.2: Onset of various behaviors precursors of tool use. Figure reprinted from Guerin et al. (2013) with permission from Frank Guerin, © 2013 IEEE.

individually (Zelazo and Kearsley, 1980). Using a rake or a stick to retrieve an out-of-reach object is a prototypical tool-use behavior, acquired around the second year of life, depending on the complexity of the task: it has been observed at 15-18 months for a medium-sized stick (Brown, 1990), or at about 3 years if a complex sequence of movements is required with a long stick (Uzgiris and Hunt, 1975).

It is important to note that those three categories do not occur as a sequence with sharp boundaries, but rather as a smooth evolution of the proportion of behaviors observed in the first 2 years of life. Fig. 2.2, reused with permission from Guerin et al. (2013), shows the onset of many behaviors of those three categories.

Also, there are dependencies between behaviors as the learning of one behavior can facilitate the discovery of other behaviors. For instance, the behavior of shaking an object in a hand can lead to the discovery of the behavior of banging an object on a surface which makes noise (Guerin et al., 2013). Babies seem to learn that some objects make sounds when banged on a hard surfaces whereas not on soft surfaces at about 10 months (Bourgeois et al., 2005). Fig. 2.2 shows the transfer of learning between some behaviors, through arrows from a behavior to another.

One of the earliest tool-use behaviors is spoon self-feeding. Spoon self-feeding

is composed of several actions: grasping the spoon, moving the spoon to the dish, loading the spoon with food, transporting the spoon loaded with food to the mouth without spillage, and emptying the food into the mouth (Connolly and Dalgleish, 1989). In a longitudinal study in the second year of life, Connolly and Dalgleish (1989) document the different types of spoon grasps and patterns of actions with the spoon and food during self-feeding. They show how this behavior improves over the months, becoming more consistent, smoother, more direct and faster, with the use of the preferred hand, of fewer grasp patterns, and with a better visual monitoring.

Many non-human animals are also able to create and use tools, as a weapon or to help feeding (Alcock, 1972). For instance, chimpanzees and New Caledonian crows can fashion a stick-like tool with the right properties from their environments in order to “fish” termites or ants (Seed and Byrne, 2010). Those behaviors can have a genetic influence, but they can also be culturally transmitted (Kenward et al., 2006). The use of a tool to act on another tool has also been observed in non-human primates (Mulcahy et al., 2005) and in New Caledonian crows (Wimpenny et al., 2009).

2.2.3 Tool-Use Learning Mechanisms

An active exploration and play with many repeated cycles of perception and action has been documented in the learning of reaching in the first year of life (Williams et al., 2015), which is one of the precursors of tool use. In tool-use tasks, the observation of play and learning has been the topic of several studies around the second year of life, either in lab session or at home, inside one session or longitudinally across sessions.

Let’s first go back to the observation of Piaget’s daughter Jacqueline from age 9 to 12 months (see Introduction). Jacqueline is seated in front of an out-of-reach toy which is placed on a coverlet. In order to retrieve the salient toy, the baby has to pull the coverlet. This task can fall into the tool-use category if we consider that the coverlet is a real tool, but does not fit all tool-use definitions as the coverlet is already physically connected to the toy. One striking observation of Piaget is the accidental character of the discovery at 9 months and the rediscovery at 11 months old: “After several failures she grasps the coverlet abruptly, which shakes the duck” and “In the course of the movements she makes to catch the object, she accidentally moves the coverlet which shakes the duck”. It seems that in the first occurrence, the baby was bored of trying to catch the toy and started exploring another object, the coverlet, which accidentally made the toy move. In the second occurrence, the baby was exploring the inefficient strategy to catch the toy, going directly with the hand, which accidentally made the coverlet move, resulting in moving the toy. The exploration of a non-tool-use strategy or of the tool object thus allowed to discover information on how to use the tool to retrieve the toy. In both cases, the baby seemed to understand the connection as she subsequently pulled the coverlet and grasped the duck. However, learning and remembering the efficient tool-use strategy do not necessarily happen directly from one success: after the first discovery at 9 months

old, the behavior is not observed after another accidental discovery at 11 months old. This behavior is rather learned from many repetitions along the weeks, such that the behavior has become systematic at 12 months.

Later in development, the repetition behaviors seem to be more variable and to allow the learning of new skills. Indeed, in the fifth stage of sensorimotor development (around 12-18 months, Piaget (1952)), Piaget emphasizes the behavior of modifying a previous experiments and observing the corresponding results: “When the child repeats the movements which led him to the interesting result, he no longer repeats them just as they are but gradates and varies them, in such a way as to discover fluctuations in the result”. One instance of this is the discovery of the tool function of the stick by his daughter Lucienne at 14 months in observation 157: “While playing at hitting a pail with a stick she is holding (all this without preliminary goals) she sees the pail move at each blow and then tries to displace the object. She strikes it more or less obliquely to augment the movement and does this many times”.

Another important learning mechanism that seems to be highly used by children notably in the context of tool use is analogy and transfer, however there is little data on that topic (Brown, 1990; Guerin et al., 2014). Through analogy, a child is able to transfer the skills learned in a particular situation (called source), to a different yet analogous situation (called target). In the context of tool use, the difference between the source and target situations can be the color and texture of the objects, the shape of the tool, the relation between the tool and the object acted upon. If the difference is small enough, a behavior working in the source situation will work in the target situation, though some exploration and adaptation might be required. Piaget reported several anecdotes of analogy, such as with Jacqueline in observation 160 (Piaget, 1952) who transfers the use of a stick as a tool to the use of a book and a banana for the same purpose. Beck et al. (2014) studied the ability to transfer tool-making knowledge in children aged 4-7 years. They show that children were able to transfer tool making to new situations when the tool could be made with the same materials and with similar shape. They argue that transfer abilities depend on memory and analogical reasoning and thus improve with age. See Guerin et al. (2014) for other examples and a discussion of analogy and transfer. In animals, analogy and transfer are also thought to play a central role in tool-use tasks. Taylor et al. (2007) argue that New Caledonian were able to solve meta-tool-use tasks in their experiments through analogical reasoning.

Play, exploration and transfer are essential learning mechanisms for tool use when no social guidance is provided. When a caregiver can guide the child, another fundamental opportunity for learning is the observation of caregiver’s demonstration and subsequent imitation attempts. Observational learning has been shown to function as early as 12 months in a task where music was produced through the bimanual manipulation of a rolling drum (Fagard and Lockman, 2010). Concerning tool use, observational learning appears later in development (Fagard et al., 2016). In a toy retrieving task, Chen et al. (2000) show that the successful use of a tool

after a demonstration can be observed from 18 months, but many children do not show observational learning before 35 months. In a similar tool-use settings with a non-salient reachable rake tool placed near a very salient out-of-reach toy, Rat-Fischer et al. (2012) show that infants start to benefit from demonstrations of how to retrieve the toy with the rake only from 18 months. Related studies revealed that learning by demonstration can be improved by showing the intention of the experimenter prior to demonstration, through trying to grab the toy and saying “I can’t get it” (Esseily et al., 2013), by implicitly repeating the demonstration, without any verbal comments (Somogyi et al., 2015) and by making the baby laugh during the demonstration (Esseily et al., 2016). In a study of tool making in 3-5 year-old children, Beck et al. (2011) tested their ability to make a hook tool adapted to solving a particular toy retrieving task, either on their own or after a demonstration. They show that the manufacture of a hook tool was easy after demonstration while the innovation of a novel tool without demonstration was difficult at this age.

The complexity of tool use, and therefore the age at which a particular tool-use behavior can be observed, depends on the properties of the relations between the tool, the object acted upon and the child sensorimotor capabilities. The concept of affordances, described by Gibson (1979), represents the relations between an object perceived in the environment and a subject, including the ways the object can be moved or acted upon, from the point of view of the subject and given its sensorimotor capabilities and constraints. Tool use has been described as a continuous developmental achievement in children in which the learning of affordances plays a central role (Lockman, 2000). Van Leeuwen et al. (1994) describe tools in terms of higher order affordance structures, which complexity depends on the interrelations between the three dual relations between the actor, the target and the tool. They show that the difficulty of hook tool-use tasks with children between 9 months and 4 years old is in accordance with the complexity of the higher order affordance structures. The authors argue that children perceive more complex affordance structures with development. In a subsequent study of hook tool use, Cox and Smitsman (2006) show that tool-use actions depend on hand preference, such that right-handed children used the hook tool as a hook to pull the toy when used with the right hand, and as a stick to sweep the toy when used with the left hand. The tool-use strategy thus depends on prior experience with using tools with both hands. Barrett et al. (2007) studied tool-use performances depending on the familiarity of the tool in 12- to 18-month-old infants. They show that using a familiar tool in a non-familiar way, such as grasping a spoon from the bowl side in order to insert the handle in a hole, makes learning harder than with a completely novel tool for a similar task. This result also suggest that prior experience with the use of objects as tools influences the perception of the object possibilities and thus shapes the object subsequent exploration.

Piaget’s theory of child development, comprising a sequence of detailed stages all children must go through, has been later criticized in several aspects, one of which is the monolithic description of thinking and sensorimotor strategies children

seem to use at any point in development and the sudden change between stages in Piaget's view. In response to this criticism and supported by more recent data, Siegler developed its overlapping waves theory (Siegler, 1996) which states that when thinking about a problem or phenomenon, children don't abruptly change their approach from one strategy to another, but rather use multiple strategies at any given point in development, which frequency of use vary gradually with cognitive development together with the introduction of more advanced strategies. For instance, in a study of tool-use development, Chen et al. (2000) experimented with 1.5- and 2.5-year-olds that had to retrieve an out-of-reach toy with one of the six available tools. Children were exposed to several problems with different tool shapes and visual features, but for each problem only one tool was effective to retrieve the toy. The authors found that 74% of toddlers used at least three strategies, mainly to lean forward and try to retrieve the toy with the hand, to grab one of the tool and try to catch the toy with the tool, to ask the mother if she could retrieve the toy for them or to walk around the table to look at the toy from different angles. They also measured the variations of strategy frequencies and argued that their results were in accordance with the overlapping waves theory.

Guerin et al. (2013) summarizes the sensorimotor development in the first two years of life as the development of two tracks: the concrete track is the set of sensorimotor behaviors, skills or schemas, and the abstract track deals with representations, built from the sensorimotor experience gained in the concrete track, and influencing the future sensorimotor exploration and learning. They list six learning mechanisms for the well-studied concrete track: repetition, variation and selection, differentiation, decomposition (in sequential chunks), composition (in sequence) and modularisation (refinement of a schema to be used as a primitive action). The synchronization between the two tracks is assumed to happen through a mechanism called "representational redescription", where abstract representations evolve through the acquisition of new data that do not fit old representations.

Overall, one consistent observation in the development of tool use and its precursors has been that babies actively explore, or *play*, and learn through this process. When the space of playable objects and behaviors is too large for the allowed time in a playing session, goal-directed behaviors have been assumed to play a role in the selection of interesting behaviors by babies once their cognitive abilities would allow them to retain a goal in memory (Guerin et al., 2013). However the goal-directed functioning of behaviors has been little studied so far, particularly in the context of tool use. Willatts (1990) argues that goal-directed behaviors in the play with external objects appear as early as 3 months where babies seem to be able to hold a complex goal in memory. The fact that a baby would want a particular goal, such as retrieving an out-of-reach toy, and keep that goal throughout the study trial has been assumed and used to infer the tool-use capabilities in many tool-use problem-solving studies. For instance, a very salient toy is placed out of reach and a non-salient tool is within reach in Rat-Fischer et al. (2012). However, babies could be choosing any other goal

that looks interesting to them because of their particular learning history, preferences for colors and shapes, current frequency use of strategies, etc.

This body of studies and theories on tool use suggests that many exploration and learning mechanisms could play a role in tool-use development, motivated by a combination of intrinsic and extrinsic motivations with social guidance, and could influence the behavior of infants and toddlers at different points in development or even interplaying in the same learning sessions.

2.3 Speech Development in Infants

2.3.1 Infant Vocalizations: from Squeals to Words

During their first year of life, infants progressively learn to manage their vocal tract to go from producing squeals, growls or quasi-vowels to producing the vowels and speech-like syllables of their native language (Oller, 2000). Fig. 2.3 illustrates the developmental progression in the first year of life on the production and perception tracks.

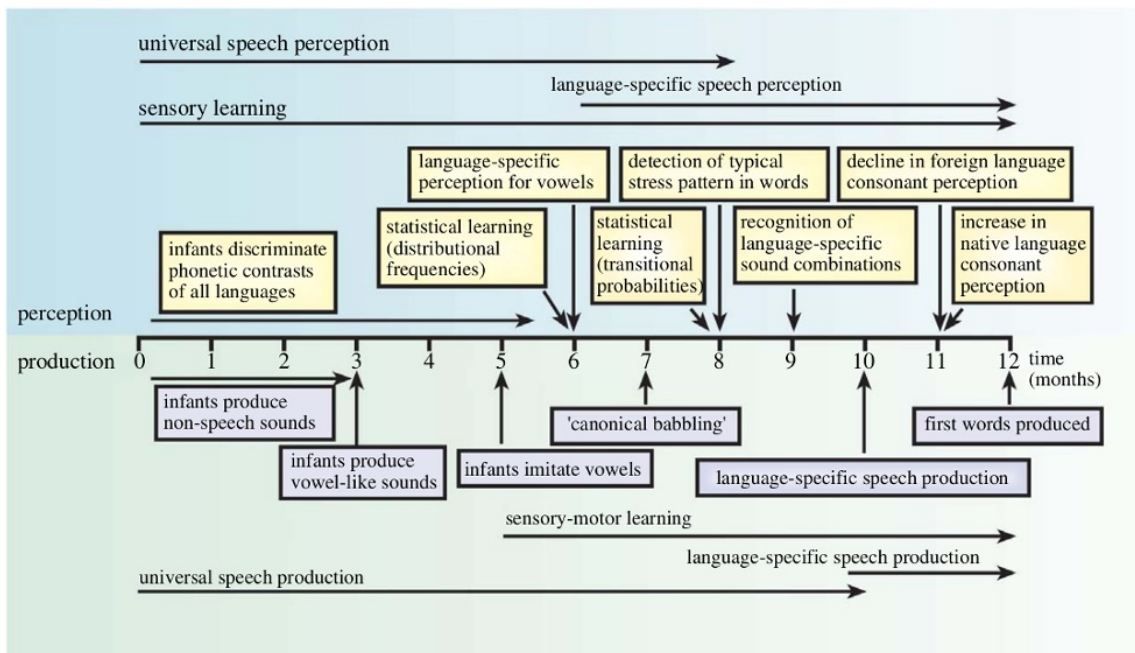


Figure 2.3: Timeline of the development of speech production and perception in the first year of life. Figure reprinted from Kuhl et al. (2008) with permission from Patricia K. Kuhl, Institute for Learning & Brain Sciences, University of Washington.

From birth to 2 months, babies start to produce quasi-vowels (Oller, 2000), sounds with normal phonation (unlike crying, sneezing). From 2 months to 4 months, they

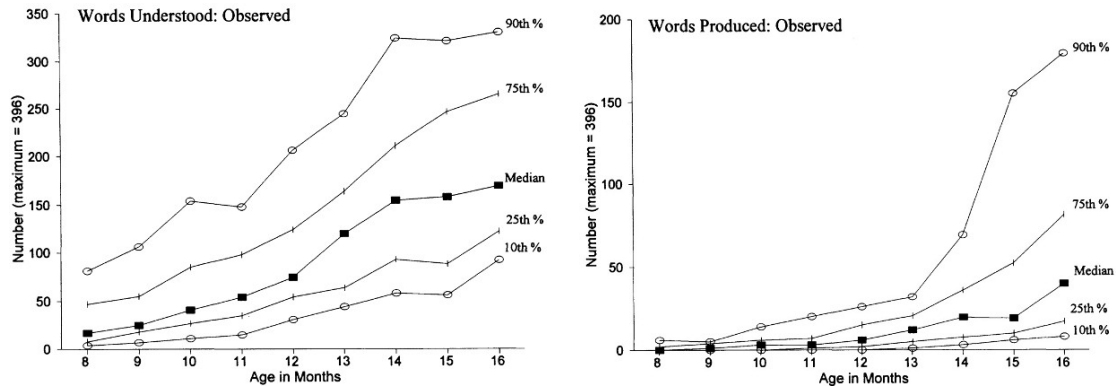


Figure 2.4: Word understanding and production abilities, with variability across infants. Figure reprinted from Fenson et al. (1994) © JSTOR.

produce articulated sounds that have been named *gooing*, with the tongue that moves during phonation which results in primitive consonant-like sounds (Zlatin, 1975). Then, from 4 months to 6 months they learn to produce fully resonating vowels. Canonical babbling is the production and repetition of syllables with one consonant and one vowel that are the building blocks of words in languages. The onset of canonical babbling is around 6-7 months. It has been shown to always appear before the production of the first spoken words in a longitudinal study (Oller et al., 1998), such that canonical babbling is considered to be critical for the learning of speech. For instance, a late onset of canonical babbling predicts developmental disorders such as autism (Lee et al., 2018). Later, by the end of the first year, speech production starts to be specific to the language learned (Kuhl et al., 2007). The first words are produced around 12 months. The first words accumulate slowly but the speed of acquisition increase dramatically in the second year of life Fenson et al. (1994). Fig. 2.4 shows the number of words understood and produced depending on age, with the variability among babies. Finally, by their third year, most children will be able to produce fully grammatical sentences Fenson et al. (1994).

During this same period, babies learn to recognize the phonemes and syllables of their language (Kuhl et al., 2007) and to understand the meaning of many words Fenson et al. (1994). For instance, babies are able in their first months to discriminate all phonetic units of all languages (Eimas et al., 1971), but this ability disappear with development in the second half of the first year, which supports the idea of a critical period for phonetic learning (Kuhl et al., 2007). Furthermore, the adaptation to the ambient sounds is predictive of normal language development: at 7 months, a better ability to discriminate the sounds of nonnative languages is correlated with reduced later languages abilities while a better ability to discriminate the sounds of the native language predicts better later language abilities (Kuhl et al., 2005).

Moreover, research on prelinguistic infants has documented that they do not learn only by passively listening the sounds of their environment and experimenting alone,

but rather by interacting with their caregiver and other adults and siblings. For instance, Gros-Louis et al. (2006) showed that mothers respond contingently to 10 month-olds' prelinguistic vocalizations 70% of the time, mainly with acknowledgment to both vowel-like sounds and consonant-vowel clusters. Furthermore, these mothers respond with more playful vocalizations to vowel-like sounds than CV clusters, and with a more imitative behavior to CV clusters than vowel-like sounds. In other words, caregiver's behavior depends on infant's vocalizations, and, in addition, infant's vocalizations properties evolve in response to caregiver's stimulation across months of development. These relations suggest an interaction loop between the prelinguistic infant and his caregiver, that might help and shape vocal learning. We describe in details such infant-caregiver interactions and their consequences on speech development in the next section.

2.3.2 Infant-Caregiver Interactions in Speech Development

Human, together with songbird, is one of the few species that learn vocalizations specific to their cultural environment (Kuhl and Meltzoff, 1996). Imitation is thought to be an important pathway to social and vocal development (Meltzoff and Warhol, 1999), constrained by the dramatic changes in the anatomy and functioning of the vocal tract in early life (Sasaki et al., 1977). Infants have been demonstrated to imitate the vowels produced by an adult speaker already at the age of 3 to 4 month-old (Kuhl, 1991). When infants are imitating their caregiver, the prelinguistic vowel categories become more separated in the vowel space from 12 to 20 week-old infants (Kuhl and Meltzoff, 1996). In response to this vocal babbling, mothers have been shown to use sensitive speech and vocal imitation, particularly when vocalizations are perceived as more speech-like (Albert et al., 2018).

In order to study the properties of the real-time interaction between the baby and the mother, and in particular how the interaction starts and sustains and how engaged and motivated the infant is in learning by this social interaction loop, Franklin et al. (2014) used a Face-to-Face/Still-Face/Reunion paradigm with 6-month olds. The mother and her child were in free interaction during the first phase to measure baseline vocalizations number and types. Then the mother was asked to stay looking at the child with no interaction during the second phase, and they were free again in the third phase. They measure an increase from the Face-to-Face to the Still-Face phase of all protophones vocalizations categories measured, full vowels, quasi-vowels, squeals and growls, but not of cry or laugh. This study shows that by 6 months of age infants have learned that they can re-engage their parent through speech-like vocalizations, which is a step toward a pragmatic use of the perlocutionary effect of their vocalizations, that will be an important component of their later communication abilities.

Goldstein and Schwade (2008) studied the importance of the contingency of the vocal responses of mothers to their 9.5-month-old infant. In a contingent group,

mothers were asked to react to their child's vocal babbling by speaking to, moving closer to, touching and smiling at their child. In a yoked control group, mothers had to respond to their child with the timing of the responses of the contingent group's mothers, so their responses were unrelated to their child's babbling. If mothers had to speak only with vowels, there was a significant increase of the percentage of infants' fully resonant vowels but not of their CV-structured syllables (consonant-vowel) with respect to the yoked control condition. If mothers had to speak only with CV structures, the increase was on the percentage of CV structures but not on fully-resonant vowels. Furthermore, there was no mimicking of the surface phonetic features like the particular vowel or particular CV structure, but rather a learning of the phonological pattern: fully resonant vowels and CV structures. This study was the first experiment manipulating in real-time mothers' vocalizations. It provides evidence that the infant is learning new vocal forms from the diversity of vocalizations in its mother's contingent speech while mimicking only would not allow it.

The difference between contingent and noncontingent feedback was also investigated in the context of nonvocal feedback to infant's babbling. Goldstein et al. (2003) studied the difference of vocal behavior of 8-month-old infants who receive contingent versus noncontingent nonvocal social feedback. The mothers were asked to smile, move towards the infant, and touch him when he was babbling. In a yoked control condition, mothers had to respond based on the contingent condition's mothers: their response were unrelated to their child's babbling. Results show that infants in the contingent interaction condition increased their proportion of vocalization with more mature voicing, syllable structure, and with a faster CV transition with respect to infants in the yoked condition, and that this change persisted in the free interaction phase after the end of the manipulation. This study provides evidence that a non-vocal social interaction mechanism can also shape babbling in real-time. It finds an important role in nonvocal feedback, suggesting that such social reinforcement could be one of the pathways of speech development. The authors highlight an ontogenetic parallel with songbirds as for instance female cowbirds do not sing but still are able to give feedback to their male chicks learning to sing.

In the previous experiments, the infant-caregiver interaction was measured during a single or a block of experimental sessions. They explored direct interactions between infants and caregivers' behaviors on a short timescale. However, those social interactions might have accumulating effects on the developmental timescale. In order to understand infant's long-term vocal development, the mother-infant free social interaction was observed in a playroom each week from 8 to 14 months in a longitudinal study (Gros-Louis et al., 2014). This study shows that overall the vowels and CV productions, the eye contact during mother-directed vocalizations (MDV), and the maternal responsiveness to object-directed child vocalizations increased with age. Longitudinal correlations show that maternal responsiveness and imitation of MDV in previous months predicts MDV in following months. Also, maternal responsiveness to MDV correlates with the difference in developmentally advanced

CV vocalizations from 8 to 14 months. These longitudinal correlations support the idea of an accumulating effect of the properties of this social interaction loop on vocal development.

Gestures and pointing behaviors have been argued to pave the way for language learning as they provide a social platform for communicating and interacting with another person while drawing attention to an object (Goldin-Meadow, 2007). The pointing behavior thus represents another opportunity to learn words and their meaning in a social interaction context (McGillion et al., 2017). The first pointing gestures with the index finger extended appear around 3 months (Fogel and Hannan, 1985), however the full gesture with both the arm and the finger extended and a communicative intent from the infant emerges between 9 and 15 months (Tomasello et al., 2007). Children learn words from their pointing behaviors. For instance, the lexical items that appear in the spoken vocabulary can be predicted in a large proportion by the child’s earlier pointing gestures (Iverson and Goldin-Meadow, 2005). Also, gestures at 14 months better predict later vocabulary size than mother speech at 14 months (Rowe et al., 2008). Children can mix words and pointing to convey more complex semantics, such as verb-object when saying “eat” and pointing at a cookie (Goldin-Meadow, 2007), and mothers often “translate” those behaviors in return with complete sentences (Goldin-Meadow et al., 2007). Children could be selectively pointing to the objects they find interesting enough to communicate and learn about (Tomasello et al., 2007), which could result from intrinsic motivations, operationalized for instance by the knowledge gap hypothesis (see Sec. 2.1.1).

2.4 Links between Tool-Use and Speech Development

Tool use and language seem to require similar information processing capabilities allowing the production and perception of sequential combinations of increasing complexity, from reaching to spoon self-feeding and from phonemes to stories.

Greenfield (1991) describes how both tool use and language show a hierarchical organization, and draws a parallel between the early development of the tool-use skills and the phonetic skills in the two first years of life that seem to show the same increases in complexity around the same age. For instance, around 8 months, babies duplicate CV clusters as in “dada” on the phonology side, and on the tool-use side are able to move a spoon in and out of the mouth or the dish and repeat. Later, between 12 and 16 months, CV1CV2 clusters with one consonant and different vowels appear as in “daddy” or “baby”, while behaviors with the spoon evolve such that babies can touch the food with the spoon then touch the mouth with the spoon, which can be argued to be of a similar structure than the CV1CV2 clusters. Actions, and in particular goal-directed actions, have thus been described as following a grammar (Pastra and Aloimonos, 2012) generating hierarchically organized action sequences.

Other specific relations between language and tool-use development were investigated by Meltzoff (1988). For instance, they studied the onset of means-end behaviors and success/failure words. They show that the onset of those two cognitive skills is close in time (13 days on average), while there is a large variability between children as they can appear from 15 to 24 months. An insightful use of tools was a better predictor of the use of success/failure words than of other abilities such as object-permanence skills.

Another potential link between language and tool use is that communication can be seen as a social tool. Although not manipulating a physical object to get food, a baby saying “eat” and pointing to a cookie is producing physical sound waves that have an effect on the caregiver. As with a stick or a spoon, the baby has to learn in which contexts this social tool-use strategy works, e.g. if it has not already eaten too many cookies, and how this vocal tool functions, e.g. that saying “eat X” might have a different effect depending on X. This idea is consistent with the fact that communicative gestures have also been described as social tools. For instance, pointing has been argued by Tomasello et al. (2007) to be used to affect caregiver’s mental states. Also, deaf children use language gestures as tools, for instance to get others to do things for them (Goldin-Meadow, 2007), when hearing children would have used sentences. Those previous studies support the idea that there could be specific relations between the cognitive processes involved in tool use and language development.

In addition to showing similarities in hierarchical organization and potentially in the required cognitive information processes, language and tool use might share some neural correlates. A first link between hand gestures and speech production supports the idea of related neural substrates between speech and hand gestures. Gentilucci et al. (2001) shows that when human subjects are asked to open grasp an object and open the mouth, the lip aperture and aperture velocity are higher when the grasped object is large than when it is small. They also show that if the subjects have to pronounce a syllable, the production of the syllable is also influenced by a parallel grasping movement. In the case where grasping movements are not executed but observed, they have also been shown to influence speech production: lip aperture and voice amplitude were higher when the observed grasped object were large Gentilucci (2003). These behaviors are thought to involve the mirror neuron system (Rizzolatti and Craighero, 2004) where neural cells have been shown to both respond if an action is observed in others and respond when that action is executed by the subject. In a work more specific to tool use, Higuchi et al. (2009) studied the brain activations during language and tool tasks in human adults with functional MRI. They found an overlap of activity in both tasks in the dorsal part of area BA44 in Broca’s area. This region has previously been reported to be used in complex hierarchical sequential processing in language, such as embedded sentences, but not for sentences without hierarchical structure (Sakai, 2005). Those results support the idea that those complex hierarchical structures, present both in tool use and language,



Figure 2.5: Left: Oldowan stone flake. Right: Acheulean biface. Picture by José-Manuel Benito Álvarez (CC BY-SA 2.5).

are processed by the same neural circuits. Furthermore, the authors argue that the ability for processing of hierarchically organized behaviors was present in our common ancestors with primates for tool use and was later exapted to support language in humans.

Those common neural correlates could have evolved in the hominid lineage, where a selection pressure for complex tool use, language and social behaviors might have together driven the increase in brain neural capabilities (Greenfield, 1991; Higuchi et al., 2009; Morgan et al., 2015).

Greenfield (1991) outlines several possible evolutionary scenarios for tool use and language in primates and humans. She proposes that the common ancestor of humans and today's primates had the neural circuitry in the left frontal lobe to support both primitive object combinations and primitive language functions, and that they evolved together in the human lineage. Better tool-use abilities would have increased the adaptive value of proto-linguistic communications, and vice versa, both would have evolved through mutually reinforced natural selection. The adaptiveness of language and tool use would have driven the expansion of the prefrontal cortex in another co-evolutionary loop.

Two and a half millions years ago, stone age's hominins were producing sharp flakes through striking a cobble core with a hammerstone (Morgan et al., 2015), and those sharp flakes were then used as cutting tools, e.g. for butchering. This Oldowan technology was geographically spread and continuously used with little changes for 700,000 years, before the advent of the Acheulean technology including more complex and diverse hand-axe tools (see examples in Fig. 2.5). The Oldowan stone knapping skill is thought to be culturally transmitted as there seem to be regional traditions. Experiments with the transmission of this skill in modern humans show that imitation/emulation was a low-fidelity transmission mechanism while teaching and language improved transmission (Morgan et al., 2015). The authors argue that imitation could have been the mechanism of tool making cultural transmission in the

Oldowan period, while teaching and proto-language could have been prerequisites for the transmission of the Acheulean technology. If tool making and its efficient transmission had an increased evolutionary fitness in the Oldowan culture of the early hominin ecological niche, a teaching and proto-language ability allowing the development of the Acheulean tool-making culture could be the result of a long gene-culture co-evolution.

Iriki and Taoka (2012) propose that language and tool-use cognitive abilities evolved from the computational processes involved in the control of reaching actions. The authors describe the interdependencies between the ecological, neural and cognitive niches for the human lineage, together called a *triadic* niche. Reaching actions, in particular in the context of bipedalism, imposed a high demand on multi-sensory integration and complex coordinate transformation, that selected brains with improved neural circuitry for processing them. In turn, those neural capabilities could have been reused for other cognitive processes such as the processing of simple tool use and proto-language, which improved the evolutionary fitness of hominins. The development of tool use and language modified the ecological niche which then selected for more efficient neural circuits. The co-evolution of the ecological, neural and cognitive niches could have slowly enabled and improved higher cognitive functions like tool use and language.

2.5 Intrinsic Rewards and Motivations in Artificial Agents

Psychological research on intrinsic motivations in humans has recently inspired many computational implementations of intrinsic motivations or curiosity in artificial agents. Artificial agents are algorithms that can choose actions to execute in an environment, real or simulated, and somehow observe the properties of this environment. When those agents are given capabilities to learn from their actions and observations, this process is called “active learning”, as opposed to other forms of learning where acting is not fundamental, such as when the observations are all already available in a dataset. The motivations for studying artificial curiosity range from the desire to model particular aspects of human or animal learning, in which case the goal is to obtain an agent that behaves as closely as the target, to the desire to improve the performance of some machines or robots to solve particular tasks where extrinsic motivations and other forms of external guidance are not enough.

Oudeyer and Kaplan (2007) explored and classified previous implementations of intrinsic motivations in artificial agents along several dimensions. One recurring aspect of those implementations is the fact that the intrinsic motivation is operationalized as a measured/computed signal that represents how much actions, behaviors or outcomes are motivating or triggering the curiosity of the agent. Those signals, sometimes called “intrinsic rewards”, are computed based on the result of previous actions in

the environment, and are used by the agent in the selection of future actions. The two main categories of this classification are knowledge-based and competence-based intrinsic motivations.

2.5.1 Knowledge-Based Intrinsic Motivations

With knowledge-based intrinsic motivations, the agent monitors the new observations in the environment and compares the new acquired knowledge with previous knowledge, such as facts, situations, objects, places, etc. They include intrinsic attractions to novelty, where the intrinsic reward would be the novelty of the observation, leading the agent to search for situations with high novelty, such as in Huang and Weng (2004). This idea has been used in Benureau and Oudeyer (2016) in order for a robot to find the maximum diversity of behaviors. In the context of Reinforcement Learning, Strehl and Littman (2008) added an exploration bonus to the Bellman recursive equations such that states that have been less visited in the past should be more visited in the future. However, such “count-based” methods are not useful in large domains where states are rarely visited more than once. Bellemare et al. (2016) used a generalization of counts based on the information gain of a model of the density of visits, called a ‘pseudo-count’. They approximate the state action value function with a Deep Q network with the exploration bonus, and show that it improves dramatically the exploration of agents in the Montezuma Revenge Atari game, one of the hardest Atari games where ϵ -greedy approaches fail.

Motivations for situations with cognitive dissonance or prediction errors are also knowledge-based intrinsic motivations. In those implementations, the outcomes of some actions are compared to the predicted outcome, such that the situations where there is a high prediction error may be good opportunities for learning and are thus sought. One example of this mechanism is found in Chentanez et al. (2005) where they use the framework of “options” (Sutton et al., 1999) and define the intrinsic rewards of salient events as the error in prediction of the events based on a learned option model for the events. In continuous environments but with discrete actions, Metzen and Kirchner (2013) also use the option framework to learn skill hierarchies with an intrinsic motivation rewarding positively the novelty of the encountered states and negatively the prediction error of the learned skill model.

Monitoring the average progress of prediction errors, a form of “learning progress” also falls into this category. For instance, the agents in Oudeyer et al. (2007) use the Intelligent Adaptive Curiosity (IAC) algorithm, preferring regions where prediction errors are decreasing on average, indicating that learning is happening. Agents in Schmidhuber (1991a) compare the prediction with the new prediction after updating the predictor to measure learning progress. In Mugan and Kuipers (2009), agents first learn a qualitative representation of environment states and actions before learning Dynamic Bayesian Networks representing the temporal contingencies of those states and actions. The authors use the IAC algorithm (Oudeyer et al., 2007) to choose

actions estimated to yield a high prediction error progress.

Another form of knowledge-based intrinsic motivations rewards actions or behaviors leading to a gain in information. Frank et al. (2014) implemented a particular reinforcement learning agent in a humanoid robot, with a low-level control layer and a high-level curiosity mechanism maximizing the information gain measured as the KL divergence between the distributions of state-action policy before and after the update from an experiment. In finite environments represented by a factored Markov Decision Process, Vigorito and Barto (2010) used an intrinsic motivation towards actions maximizing the learning of the structure of the environment, which improved learning efficiency compared to the case without the intrinsic motivation. In Houthoofd et al. (2016), the Variational Information Maximizing Exploration (VIME) exploration strategy is based on the maximization of the information gain of the agent’s model of the environment. They report an improvement of exploration performances in a variety of benchmark continuous tasks, such as MountainCar, HalfCheetah or SwimmerGather.

2.5.2 Competence-Based Intrinsic Motivations and Goals

Agents with competence-based intrinsic motivations consider their skills to solve tasks or goals. Goals have been introduced in machine learning in Kaelbling (1993), where they represented states in a Markov decision process. Since then, goals have represented several concepts related to artificial agents and their environment, such as particular states to reach, outcomes to realize, behaviors to show, objects to affect, etc. The notion of goal-directed behaviors has seen a surge of interest in the last decade in developmental robotics and artificial intelligence research. Central questions in this area have concerned the possibility for an artificial agent to discover, represent, evaluate, select or generate goals with a potential for learning.

Within the Reinforcement Learning framework, several algorithms were designed to find interesting goals and subgoals in a context where the reward function already defines an overall task in the environment. In Stout and Barto (2010), the agent chooses the skills to train, with an intrinsic motivation for the ones showing competence progress, whereas skills already learned or too difficult are not chosen. In that study, the skills are drawn from a predefined set, and involve actions and observations in a discrete world. In Schaul et al. (2015), the value function is extended to include a goal parameter in a universal value function approximator (UVFA). In Dosovitskiy and Koltun (2016), goals are predefined combinations of future measurements (such as ammo, health and frags in the Doom video game), and agents that learned with a range of random goals generalize better than models learned with a fixed goal.

In Hierarchical Reinforcement Learning (HRL), the “option” framework (Sutton et al., 1999) proposes to represent temporally-extended actions, called options, through semi Markov decision processes with policies having a particular initiation state set and termination condition which can depend on history. Those policies can then be

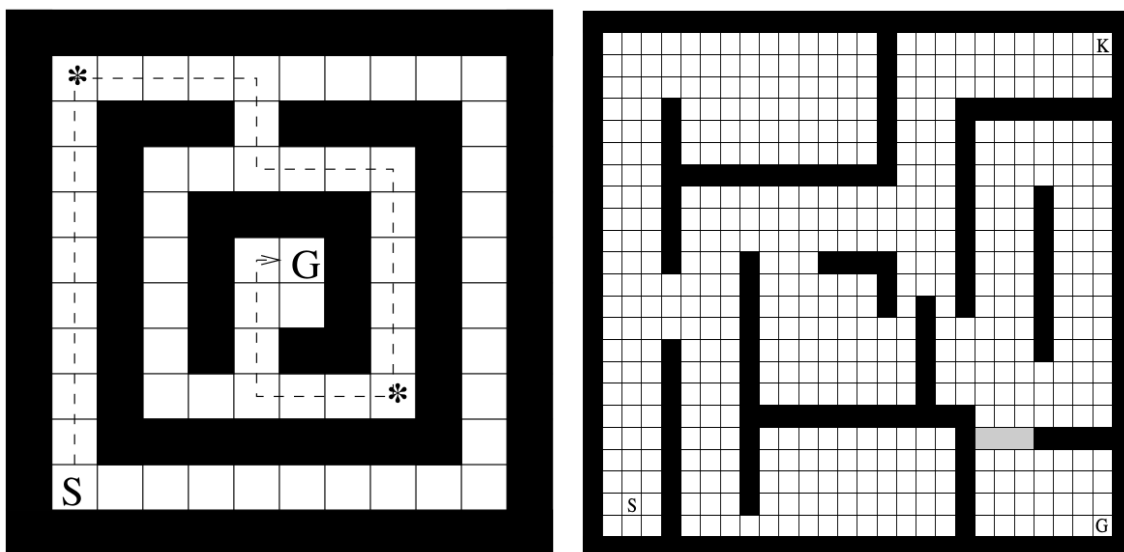


Figure 2.6: Two mazes used for testing hierarchical reinforcement learning agents in (Wiering and Schmidhuber, 1996), reprinted with permission from Marco Wiering. Left: a simple maze, with ambiguous positions. Right: the key K must be found to be able to cross the door leading to the goal G .

used as primitive actions in the traditional learning of a Markov decision process representing the problem. Several propositions have been made to identify subgoals to be used as options. McGovern and Barto (2001) defined subgoals as states where the agent go back regularly in successful trajectories, leading to the identification of “diverse density regions”. However, with this definition subgoals can’t appear in non-successful trajectories. Many variants of this bottleneck idea have been used, such as the notion of betweenness in a graph, identifying the states common to many shortest paths between states (Şimşek and Barto, 2009), or the concept of relative novelty, allowing to recognize states used to transition to novel regions (Şimşek and Barto, 2004), or others (Goel and Huber, 2003; Kretchmar et al., 2003). With an information-theoretic approach, Van Dijk and Polani (2011) proposes to measure the amount of Shannon information that the agent needs to maintain about the current goal at a given state to select the appropriate action, and to identify distinct information transition states as subgoals. Other approaches use a clustering of states into regions to then identify subgoals as transitions between regions (Bakker et al., 2004; Entezari et al., 2010; Mannor et al., 2004; Menache et al., 2002; Şimşek et al., 2005). Many of these studies report improvements of the learning efficiency compared to agents without a goal hierarchy, however they mostly consider toy problems in small and discrete gridworlds. For instance, Wiering and Schmidhuber (1996) use the mazes as reprinted in Fig. 2.6. Their hierarchical extension of Q-learning with subagents learning Markovian subtasks succeeds to reach the goals while Q-learning

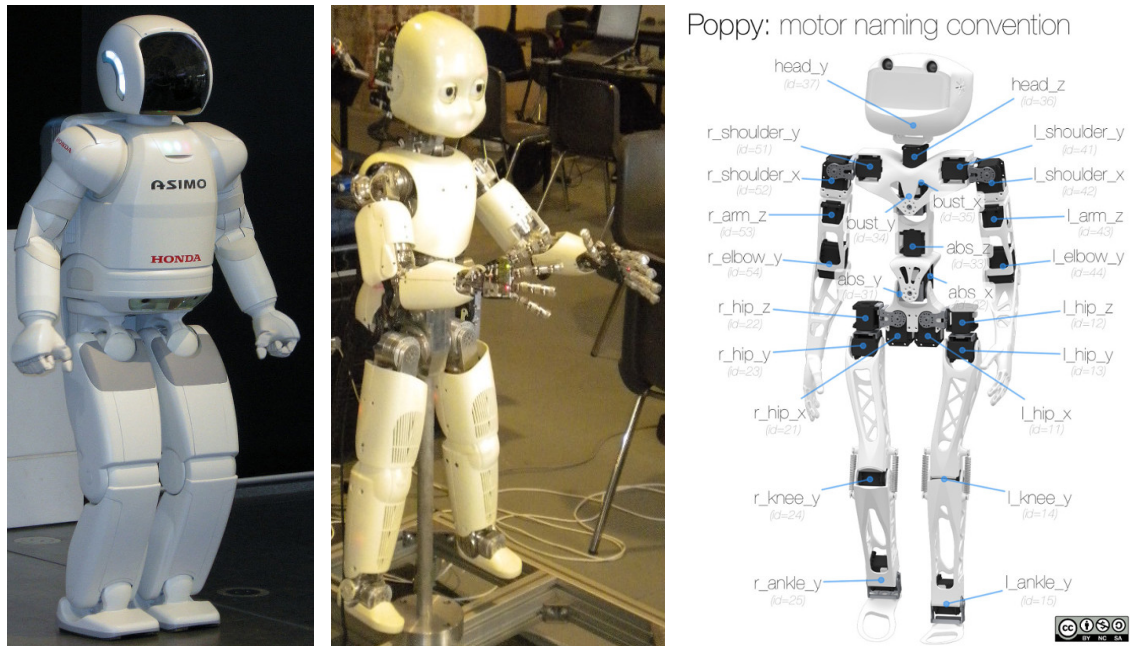


Figure 2.7: Left: Honda Asimo robot. Middle: iCub humanoid robot (attribution: Jll at English Wikipedia). Right: Poppy Humanoid robot (attribution: poppy-project.org).

fails. In Gregor et al. (2016), instead of looking for a small number of options that can be reused as subgoals, the agents learn all possible “intrinsic options”, options that affect the world in a meaningful way, through learning an embedding space for options. In Kulkarni et al. (2016), a hierarchical DQN is used to learn a top-level policy over intrinsic goals and a lower-level policy on atomic actions to reach goals.

In the field of developmental robotics, learning agents are usually embodied in a robot with a continuous space of actions and a continuous space of observations, which can both be of high dimension, and are given limited exploration time given robotics constraints. For instance, a humanoid robot can contain 50 motors each of which is typically controllable continuously between two angular bounds (see examples in Fig. 2.7), with an action space containing as many degrees of freedom (DOF), and a stream of observation coming from a camera with many pixels or from a pre-processed vector with many features representing the state of the environment. Therefore, the reinforcement learning algorithms previously discussed are not adapted to learn in such a setting.

In developmental robotics settings, a typical task the agent is facing is to learn the functioning of its body through experimentation. A first approach to discover the relation between the motors and the body parts is to move all joints in a random manner and observe the effects on the body. However, this approach is not efficient to produce diverse behaviors with a high-dimensional body as in such cases there is a

lot of redundancy in the motor actions to produce behaviors and the most interesting behaviors are produced when motors are moved in a coordinated manner unlikely to be found with random actions. Rolf et al. (2010) used a “goal babbling”, or goal-directed approach, where the agent targets its exploration to different goals and tries to reach them. They showed that goal babbling improved learning of the control of a 2D simulated arm’s end-effector compared to motor exploration, and that this approach scales to high-dimensional action spaces such as the arm of a Honda robot (see Fig. 2.7). Several reasons explain the efficiency of goal babbling compared to motor babbling, the exploration of motor actions. One is that the redundancy of high-dimensional body parts such as arms with many joints implies that many motor configurations result in the same end-effector position, such that maximizing the diversity in the exploration of motor parameters do not necessarily maximize the diversity of end-effector positions. Instead, focusing on goal positions for the end-effector increases the diversity of reached positions. Another reason is the fact that while exploring towards one particular goal, this goal being reached or not, there are many chances that the exploration path leads to the discovery of other skills, that can be reused when exploring other goals later on (Rolf et al., 2010). This improvement has been replicated in many other robotic setups and learning contexts, such as the goal-directed learning of hand-eye coordination in a Nao robot (Schmerling et al., 2015), the learning of the movement of an arm to throw a ball in a socially-guided context (Nguyen and Oudeyer, 2012), or the control of a quadruped robot (Baranes and Oudeyer, 2013). By comparing the speed of learning of several goal tasks in a 2D simulated robotic arm depending on the intrinsic motivation signal, Santucci et al. (2013) shows that a knowledge-based intrinsic motivation signals such as prediction error or prediction error improvement are inadequate for the learning of multiple skills, namely reaching different objects with the arm’s end-effector. Instead, competence-based intrinsic motivations taking into account the current competence of the agent are able to learn in such settings.

In the process of learning to reach goals, agents are producing a diversity of behaviors in the goal space. Therefore, goal-directed learning has similarities with other learning approaches that push the agent to find novel or diverse points in a particular space, such as in “novelty search” (Lehman and Stanley, 2011a; Pathak et al., 2017) and “quality diversity” algorithms (Cully et al., 2015; Cully and Demiris, 2017). Churchill and Fernando (2014) define a cognitive architecture for the control of a humanoid robot made of a graph of operation and goal nodes, that is evolved through mutations and recombinations.

2.5.3 Intrinsic Motivations and Learning Curricula

Although little research has been carried out on the developmental consequences of intrinsic motivations in babies, the parent’s point of view tells that babies seem to be able to some extent to choose their sequence of learning activities, or learning

curriculum. For instance, in their first year of life, babies train and learn to control their head and trunk, to roll their body, to sit, to crawl, to stand up, with approximately the same developmental sequence across babies, although with differences in timings and sometimes in the order of stages, such as standing up before crawling. In this stages and especially in the earliest ones, few guidance is provided, which boils down to putting the baby in positions where it has the opportunity to safely discover and experiment the next skills. In this same period, babies are undergoing tremendous maturational changes in their body shape and dynamics and in their cognitive abilities. Important preliminary results showing that neural networks may learn more efficiently if starting with smaller data and memory constraints (Elman, 1993), together with the hypothesis that intrinsic motivations might have some responsibility in the development of many skills in the first years, have driven some focus on the relation between computational implementations of intrinsic motivations and the autonomous emergence of a learning curriculum.

In the Playground experiment (Oudeyer et al., 2007), a quadruped robot is placed in an infant play mat, with a contingent robot peer next to it (see Fig. 2.8). The agent has to learn how to use its motor primitives to interact with its environment (with the IAC architecture). The authors observe the self-organization of developmental trajectories: the robot explores objects and actions with increasing complexity. For instance the quadruped robot shows non-affordant bashing and biting behaviors (trying to bite objects that can not be) before the affordant bashing and biting behaviors, and biting before bashing as the dimensionality of the bashing behavior is higher in this setup. They argue that these developmental trajectories have a similar structure across different runs but also some individual differences, which is consistent with what can be observed in the learning curriculum of human children across development.

Lopes and Oudeyer (2012) described a learning framework where a student can train on multiple topics, each of which has a learning curve describing the score of the agent depending on its training experience. The goal of the student is, given a constraint on training time, to optimize its allocation of time on the different topics in order to maximize its average score the day of the exam. They show that a greedy maximization of learning progress with a multi-arm bandit is an optimal learning algorithm in settings where topics have a particular learning curve (submodular: a training experience improves the score more if it happens earlier). This approach is called the strategic student approach, and it is general in the sense that it makes no assumptions on the nature of the student and the topics. They argue that this framework can represent the task faced by a learning agent in a lifelong learning settings where many skills need to be learned and reused in development.

In goal babbling, learning agents use a representation of goals, either given or learned. The problem of learning to reach all represented goals can be formulated as how to choose interesting goals: how to define an intrinsic reward for goals and how to select them based on this measure. In Baranes and Oudeyer (2010a, 2013), goals

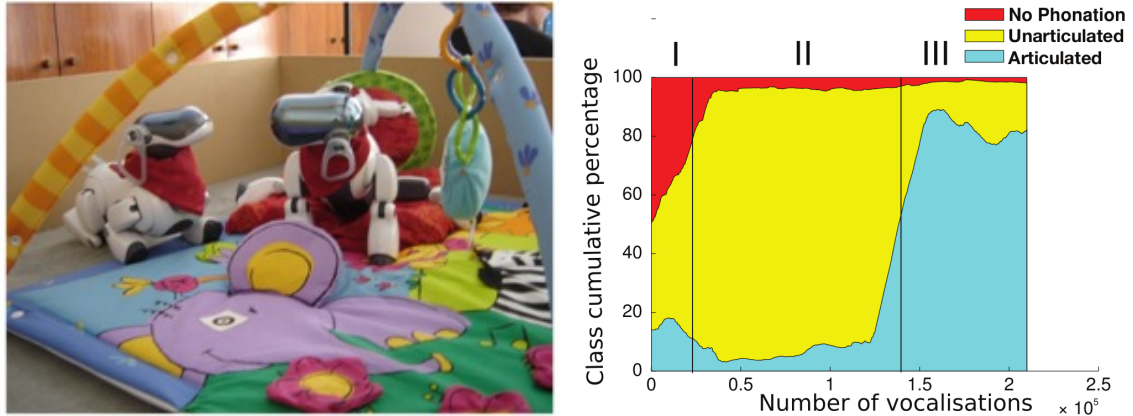


Figure 2.8: Self-organization of developmental trajectories. Left: a quadruped robot in the Playground experiment (Oudeyer et al., 2007). Right: an agent learns to vocalize with a simulated vocal tract (Moulin-Frier et al., 2013).

are represented and selected through the Self-Adaptive Goal Generation - Robust Intelligent Adaptive Curiosity (SAGG-RIAC) algorithm. This algorithm monitors the learning progress made to reach goals in different regions of a continuous goal space, and hierarchically divides this space to separate the regions with a different learning progress. The competence of the agent to reach a goal is defined as the negative distance between the goal and the reached point, while the learning progress in a region is computed as the absolute value of the derivative over time of the competence to reach points in that region. Those regions and their associated learning progress then serve as a basis for the selection of new goals, by first choosing a region with a high learning progress, then choosing a random goal in that region. They evaluate this intrinsically motivated agent in several environments including a simulated arm, a quadruped robot and a fishing rod experiment. The authors show that the active selection of goals based on learning progress in the SAGG-RIAC algorithmic architecture improves learning efficiency in the simulated arm and the quadruped setups, compared to a variant with a completely random choice of goals, called SAGG-RANDOM.

When a learning robot is subject to maturational constraints on its morphology and abilities, intrinsic motivations have been shown to interact smoothly with those maturational constraints (Baranes and Oudeyer, 2010b). With a simulated robotic arm constrained by a maturational clock, the McSAGG architecture focuses progressively on goal areas that are newly accessible due to the advancement of the maturational clock. This idea has also been combined with the SAGG-RIAC algorithm, leading to the McSAGG-RIAC architecture (Baranes and Oudeyer, 2011) and evaluated on a 12 DOF simulated quadruped robot. They show that the coupling of intrinsic motivations and maturational constraints through bidirectional interactions allows the progressive and efficient developmental learning of inverse models in high-dimensional

robots.

Nguyen and Oudeyer (2012) combine autonomous exploration with interactive learning, where the agent can mimic and emulate demonstrations of a peer. In their architecture, called Socially Guided Intrinsic Motivation with Active Choice of Teacher and Strategy (SGIM-ACTS), the learner actively and hierarchically chooses what to learn, how to learn, and from which teacher in case of learning by demonstration. They demonstrate the benefits of such a combination of intrinsic motivations with social guidance in a robotic setup with a simulated robot that has to throw a ball, compared to self-exploration only (SAGG-RIAC), mimicry only, emulation only and other variants.

In Moulin-Frier et al. (2013), the SGIM-ACTS architecture allows a simulated agent to understand how to use a vocal synthesizer with the help of humans' phonetic items. Between self-exploration and the imitation of human sounds, the learner chooses the strategy that shows the best competence progress. When self-exploring, the agent is generating phonetic goals to reach with the simulated vocal tract, in parts of the sensory space where competence progress is high, based on a Gaussian mixture model of the joint sensory, competence, and time space. The authors show that developmental trajectories of increasing complexity are emerging, with regularities and diversity. In a first stage, for about 30k vocalizations, the agents produce mainly unarticulated vocalizations or no phonation. In a second stage, until approximately 150k vocalizations, they produce mainly sounds that start with one vowel, and in a third stage, they produce mainly articulated sounds: VV, CV, VC (see Fig. 2.8). The diversity comes from different mechanisms: random generation in the algorithms, variability in the environment, and the multiples attractors of the learning dynamical system.

In Fabisch and Metzen (2014), the agents learn in a setting with a discrete goal space (called contexts) with a Multi-Armed Bandit algorithm (D-UCB) to choose on which goal they should train. The authors also show that learning is more efficient with intrinsic rewards based on the learning progress than a random choice of goal, in a task where a simulated robotic arm has to throw a ball at different goal places. Other work related to “policy search” study the learning of parameterized skills, with a model (Kupcsik et al., 2017), or through bootstrapping techniques (Queißer et al., 2016).

In the framework of options, Kompella et al. (2017) use an intrinsic motivation for learning progress, with a sensory input directly from pixels of a camera. The agent learns a compact set of low-dimensional representations of the pixel stream through incremental slow feature analysis. Skills include learned actions and a learned slow feature representation. The authors experiment reaching and grasping skills with the high-dimensional humanoid iCub robot. They show that skill acquisition is continual: the knowledge acquired in training one skill (e.g. the topple skill) is reused to learn other skills (grasping skill). However in these experiments, the agent do not learn arm movements: a task-relevant map is already given to the agent containing 6 actions

(move hand in several directions, open and close hand). This is done to avoid the complexity of learning to use the 41 DOFs of the iCub upper body.

Bengio et al. (2009) coined the term “curriculum learning” to denote the process of learning following a particular curriculum, with the idea that it could guide the optimization process, “either to converge faster, or more importantly, to guide the learner towards better local minima”. In their experiments, they demonstrate an improvement in learning when the network is trained first with the easy samples and later with the more complex samples, where the easy samples are ones with less noise in a regression task, or shapes with a simpler geometry in a visual categorization task. The learning curriculum is thus designed by an expert human who knows which are the easiest learning situations.

A hand-designed curriculum has also been used in Zaremba and Sutskever (2014) where recurrent neural networks with Long Short-Term Memory units learn to evaluate short computer programs such as the addition of two integers. On this task, a curriculum learning based on a mix of problems of random complexity and of increasing complexity (controlling the length of integers and the number of time they can be combined in operations), is shown to improve learning compared to only samples of increasing complexity or of random complexity.

Graves et al. (2017) combined curriculum learning with knowledge-based intrinsic motivations in neural networks. They used an intrinsic reward signal based on the gain in prediction on one hand, and the gain in complexity of the neural network model in other experiments. The learning agent is able to choose the complexity of the training samples, such as the length of n-grams in a linguistic task, or the lengths of sequences and the number of copies in a repeat-copy task. The authors report improvements in learning with the use of intrinsic motivations compared to a uniform selection of tasks, but also note that this uniform sampling is a strong baseline in those experiments. They describe interesting emerging developmental trajectories such as targeting small n-grams first and gradually increasing their size, and focusing on short sequences with high repeats then long sequences with low repeats in the repeat-copy task. In Srivastava et al. (2013), the PowerPlay algorithm self-generates novel but solvable abstract tasks, such as pattern-recognition tasks. The agent solves tasks of increasing complexity while compressing the knowledge and reusing the skills learned in previous tasks.

The concept of learning “auxiliary tasks” is related to curriculum learning in the sense that tasks that are not directly related to the learning of a final task are trained throughout the curriculum. Several auxiliary tasks have been implemented and shown to improve learning in particular in the context of sparse rewards, e.g. the control of pixel changes or the control of intermediate features of a neural network used for policy or value prediction (Jaderberg et al., 2016).

2.6 Tool-Use and Speech Learning in Artificial Agents

In a seminal review of state-of-the-art developmental robotics achievements and challenges around action and language learning, Cangelosi et al. (2010) suggested that the study of embodied cognitive agents, in particular humanoid robots, can help us understand the processes of sensorimotor, language, and social development in humans, and subsequently improve the learning and communication capabilities of cognitive robots. Key challenges included the learning and representation of compositional actions and lexicons, the learning in social interaction, and the codevelopment of action and grounded language in an integrated framework. In the following sections we review previous computational and robotic models of tool use and language learning, with a particular focus on the early processes of their development: learning the first movements and vocalizations, understanding that an object can be used as a tool, and that speech sounds can have effects and meanings.

2.6.1 Robotic Learning of Tool Use

The learning of affordances of objects can be seen as a necessary phase of tool-use development, the second stage in the three-stage description of tool-use development by Guerin et al. (2013): behaviors without objects, behaviors with a single object, and behaviors with object-object interactions. In this object learning phase, Ugur et al. (2015) set up several stages for a robotic learner: the experimentation of predefined behavior primitives and the corresponding discovery of tactile feedback, the learning of the detection and prediction of object affordances, and the imitation learning from human movements. The robotic agent is a 16 DOF arm with tactile feedback (signaling contact with objects at fingers or hand), given high-level visual perception mechanisms (providing the size, position and shape of objects), and high-level actions primitives (push, no-touch, release, grasp) parameterized by 3 positions: initial, target and final, and the timing of opening and closing of the hand. In this setup, a closing reflex is implemented for the hand when an object is detected through tactile feedback. The authors report an improvement of the prediction of push or grasp affordances through experimentation, and a successful imitation of simple human movements (pushing an object in a direction). They also document the evolution of human demonstrations as a response to the robot not learning a more difficult task (moving an object from position A to B while avoiding an obstacle), going from smooth demonstrations to trajectories with pauses to decompose the steps. It should be noted that in this predefined learning curriculum, the phases and their transitions are specified by hand and do not adapt to the learning outcomes.

In a following work, Ugur and Piater (2016) study the intrinsically motivated learning of object affordances from a dataset of affordances. They define several

actions: stack, top-poke, side-poke, front-poke, and effects: pushed, turned, resist, nothing, stacked, inserted, etc. They fill by hand an affordance table with 83 objects and more than 7000 object-action-effect relations. The learning agent actively chooses the actions and objects to observe from this database based on intrinsic motivations: the learning progress for actions, and the novelty for objects. The authors argue that developmental trajectories autonomously emerge from this learning framework, with actions that concern only one object being learned before actions that involve two objects (such as stacking). For instance, the affordance predictors for stacking take into account lower-level affordance predictors for actions involving one object.

A first work involving actual tools is the one of Stoytchev (2005). In this study, a robotic arm is able to grasp one of several tools (sticks, L- or T-shape hooks, see Fig. 2.9) in order to move an object towards goal positions. The learning agent is given predefined motor primitives (extend arm, contract arm, slide left or right, position wrist, grasp), and can experiment them, through random exploration, on each tool while observing the results of its actions on the position of the target object. The exploration experiments are used to fill an affordance table, relating the actions, tools and results, that can be reused to solve other tasks in new situations. This “hook” task is very similar to many tool-use tasks that have been given to children in developmental psychology experiments aiming to assess their tool-use learning abilities depending on age and experience, such as Chen et al. (2000), and is also found in other robotic setups.

In Tikhanoff et al. (2013), the iCub robot is endowed with motor capabilities for reaching, grasping and pushing objects, together with perception algorithms for recognizing, from pixel images, the distances of objects, sizes of tools, and other geometric reasoning. The robot then learns to roll and pull objects with a tool and learns the affordances depending on the tool and the objects acted upon (see Fig. 2.9). In Gonçalves et al. (2014), four actions (left, right, pull closer, push away) are available to the robot, and Bayesian networks learn the effect of those actions on objects. The affordance model is learned in simulation and only tested on real robot. The agent relates the visual features of both objects and can transfer to new tool shapes, so that the authors argue that a tool concept develops from this exploration with objects. In Mar et al. (2018), the iCub robot is given a tool in the hand (hoe, hook, rake, stick or shovel), and learns the effects of predefined actions depending on the tool shape and pose, and is able to transfer those effects to new shapes and poses.

However, in all those studies, the tool is already attached to the hand by the experimenter or the robot is given grasping motor primitives, and the robot is also given predefined primitives to move the hand and the tool. Therefore, the robot do not have to learn first its arm kinematics, but more importantly that some objects are useful when used as a tool while others are not, and how to actually use an object as a tool (e.g. grasp it by the handle).

Braud et al. (2017) propose a modular architecture for learning tool use where first basic skills are learned, such as controlling one particular sensor, and then

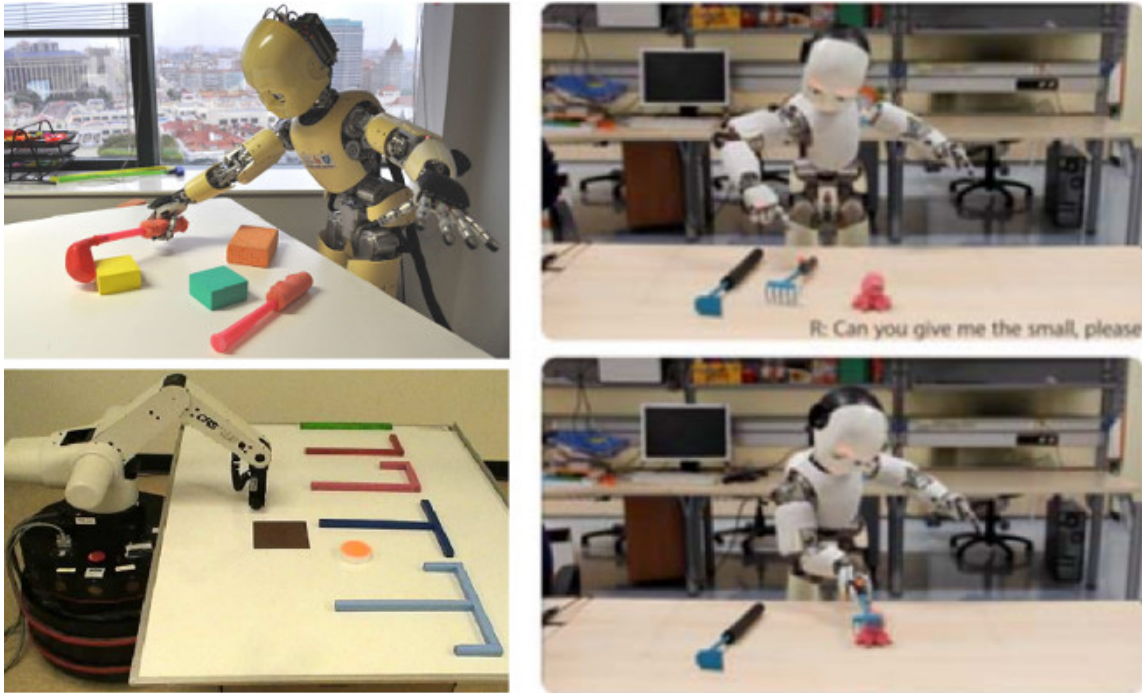


Figure 2.9: Tool-use learning in robotics setups. Top left: iCub robot (Gonçalves et al., 2014). Bottom left: CRS+ A251 manipulator (Stoytchev, 2005). Right: iCub robot (Tikhanoff et al., 2013). © IEEE.

combinations of those skills are used in sequence. In their experimental evaluations, the robot is a 4 DOF Katana arm, and 2 tools are available in the environment, which when grasped extend the arm either vertically or horizontally. The goals for the use of tools is to reach a 3D target with the tool, so the tool is not used to act on a real physical object but rather seen as just an extension of the arm with a new section.

Some research has also focused on interactive situations where the robot can learn tool use by demonstration from a human peer. In Li and Fritz (2015), the Baxter robot learns the use of electric tools such as a tacker and an drill, from kinesthetic demonstration: the teacher moves the hands of the robot to use the tool. Kinesthetic demonstrations are useful in a perspective where humans want to efficiently teach the robot new behaviors, however in a human modeling perspective, learning by kinesthetic demonstrations in humans is not the most common tool-use learning mechanism, and involves challenges that robots may not face, such as remembering the demonstrated trajectories or understanding which motor command has to be executed to reach a particular demonstrated position. In Brown and Sammut (2012) a simulated wheeled robot learns by demonstration and experimentation the use of T- or L-shaped tools that can be grasped to retrieve an object in a tube. Primitive actions are available: goto position, pickup and drop object, together with reasoning

abilities (inductive logic programming), which results in a fast learning of the use of the tool, with few experiments (between 10 and 15). In an extension of this work (Wicaksono and Sammut, 2015), the learning agent can create novel tools from satisfying constraints and running the Prolog solver as a generator of candidate solutions. The authors choose the best tool to 3D print based on their common sense among the generated candidates, and give it to the robot for use. In this experiment, the spatial properties of the tools are provided to the agent so that it does not have to learn them by demonstration.

2.6.2 Speech Learning by Computational Agents

Computational models of vocal development make use of a simulated vocal synthesizer that the learning agent must control in order to produce vocalizations, and rely on feedback from humans whose sounds can be used as targets of imitation (see Räsänen (2012) for an early review).

The DIVA model (Guenther, 2016; Guenther et al., 2006; Guenther and Vladusich, 2012) is a neural network simulating cortical interactions producing articulatory movements and receiving auditory feedback. This model provides an account of different speech phenomena, such as co-articulations. The DIVA synthesizer, which we use independently of the DIVA model in this thesis, works by specifying 10 motor parameters representing the midsagittal MRI vocal profile, the glottal tension, glottal pressure and voicing, resulting in an auditory representation encoding the formant positions (F0 to F3).

The Elija model (Howard and Messum, 2011, 2014; Messum and Howard, 2015) uses an articulatory synthesizer to produce sounds, and is rewarded for the exploration of its vocal outputs. In Howard and Messum (2014), the model also interacts with a caregiver that imitates its sounds like a mother would do: either mimicking the infant's sounds or providing an intermediate sound between the infant's one and the adult one. The computational model manages to learn object names by trying to reproduce caregiver's utterances.

In a neural network model of motor prespeech production with self-organizing maps (Warlaumont et al., 2013), a reinforcement based on the similarity of the model's output sounds with a given set of vowels biases the post-learning model's babbling sounds towards that reinforced set of vowels. In Warlaumont (2013), the reinforcement is based on the salience of the produced vocalizations, following the hypothesis that the caregiver's feedback depends on the salience of sounds.

In Moulin-Frier et al. (2013), the intrinsically motivated agent chooses the strategy that shows the best competence progress: either autonomously training to reach phonetic goals, or trying to imitate human sounds. They show that the intrinsic motivation for learning progress self-organizes coherent infant-like developmental sequences, from unarticulated sounds to consonant-vowel clusters produced through the DIVA synthesizer.

The agent of Philippsen et al. (2014) uses a recurrent neural network to learn the forward and inverse model of the VocalTractLab speech synthesizer. Their learning algorithm allows an efficient use of human supervision in the form of few examples of consonant-vowel sequences to be imitated. In Philippsen et al. (2015), they study different sensory spaces to be used as a goal space for a goal babbling agent. Instead of the traditional use of formants as sound features, they propose to use high-dimensional acoustic features based on a cochlea model, and to reduce the dimension of this space to be used as a low-dimensional goal space. A Gaussian mixture model is then used to estimate a target distribution of goals in this space, based on ambient speech sounds.

Najnin and Banerjee (2017) developed a predictive coding framework for the development of speech production and perception. Their model learns initially by self-exploration of the DIVA synthesizer and later by imitation of an ambient language, with a manual switching between the two. Random goal generation leads to the self-organization of developmental stages: from no phonation until about 10k vocalizations, to unarticulated speech until about 80k vocalizations, to articulated speech. They show that the progression through developmental stages is faster when using MFCCs as acoustic features versus formants.

Those models of language acquisition study several developmental pathways to the learning of forward and inverse models of a simulated vocal tract, from autonomous exploration to human sounds imitation. However, agents are not situated into a physical environment where vocalizations have a meaning related to objects.

2.6.3 Robotic Learning of Speech and Action

Several works study joint action and language learning, but give an advanced knowledge of the linguistic interaction protocol to the learning agent who has to associate predefined actions or objects to predefined labels and learn the semantic compositionality (Billard, 1999; Cangelosi et al., 2010; Roy, 2002).

In Sugita and Tani (2005), a wheeled robot with an arm learns to associate lexical symbols to behavioral categories through supervised learning (point, push, or hit the red, blue, green, left, center, or right object). They show that the agent is able to learn the behavioral meaning of simple combinations of words.

Dominey et al. (2009) designed a robot-human interaction scenario where the HRP-2 humanoid robot is able to understand the meaning of new linguistic instructions (such as "Give me X") by grounding them with preexisting motor skills. In this scenario, another set of predefined linguistic instructions are available to help the interaction, such as "Ok", "Wait", "Learn-macro X".

In Massera et al. (2010), a simulated robotic arm controlled by a neural network manipulates objects on a table. The neural network takes a linguistic instruction as input in the form of three values that represent the type of behavior that the robot should exhibit. They show that with linguistic inputs that guide the robot in real time towards a lifting behavior as a sequence of reaching, grasping and lifting, then

the robot learn this behavior faster than without those inputs.

In Tikhanoff et al. (2010), a simulated iCub is given a speech understanding module, a vision module, and a dataset of speech instructions, visual objects and corresponding expected actions. The robot learns from this dataset the actions to perform depending on the instruction and the available object in the scene.

To our knowledge, there is no robotic model able to learn to produce words that have a meaning in a physical environment starting from scratch through the exploration of a vocal synthesizer and possibly in interaction with a human or another robot that already knows the meaning of words.

Part I

Intrinsic Motivations in Child Development

Chapter 3

Intrinsic Motivations: Impact in Child Experiments and Role in Child Development

“Oh, no!”

— a 21-month old who solved the task

Summary

Children are so curious to explore their environment that it can be hard to focus their attention on one given activity. Many experiments in developmental psychology evaluate particular skills of children by setting up a task that the child is encouraged to solve. However, children may sometimes be following their own motivation to explore the experimental setup or other things in the environment. We suggest that considering the intrinsic motivations of children in those experiments could help us understand their role in the learning of related skills and on long-term child development. To illustrate this idea, we reanalyze and reinterpret a typical experiment aiming to evaluate particular skills in infants. In this experiment run by Lauriane Rat-Fischer et al, 21-month olds have to retrieve a toy stuck inside a tube, by inserting several blocks in sequence into the tube. In order to understand the mechanisms of the motivations of babies, we study in detail their behaviors, goals and strategies in this experiment. We show that their motivations are diverse and do not always coincide with the target goal expected and made salient by the experimenter. Intrinsically motivated exploration seems to play an important role in the observed behaviors and to interfere with the measured success rates. This new interpretation provides a motivation for studying curiosity and intrinsic motivations in robotic models.

Our study is done in collaboration with Lauriane Rat-Fischer. The author contributions were the following: LRF et al designed and run the tool-use experiment, LRF, SF and PYO had the idea of further analysis of this experiment, LRF, SF and PYO designed the ethogram, LRF and SF coded the videos, LRF, SF and PYO analyzed the results, LRF and SF wrote Section 3.1, SF wrote the other sections.

Tool use is a remarkable achievement of several species, among which several birds, primates and dolphins. They use tools to help with feeding, hunting, or building, and some of those behaviors can be culturally transmitted. In the human lineage, the oldest trace of the creation of tools dates from the stone age about 3 million years ago, when hominins were producing sharp stone flakes through striking a cobble core with a hammerstone (Morgan et al., 2015). In modern humans, the earliest tool used in infancy may be one of the sticks, spoons or rakes. Connolly and Dalgleish (1989) document the progression of different types of spoon grasps and patterns of actions with the spoon and the food during self-feeding. They show how this behavior improves over the months, becoming more consistent, smoother, more direct and faster, with the use of the preferred hand, of fewer grasp patterns, and with a better visual monitoring. However, little is known on the learning and developmental mechanisms leading to the successful use of such tools around the second year of life (Guerin et al., 2013).

Most research on the development of tool use in infancy has focused on understanding the particular tool-use capabilities that have been or can be acquired by infants at a given age. For instance, Uzgiris and Hunt (1975) show that a horizontal string task where a string must be pulled to retrieve a toy is succeeded at 12 months, while a vertical string task is at 13 months. They also show that using a medium-sized stick to retrieve a toy is displayed around 15-18 months. Bates et al. (1980) study the tool-use abilities of 10 month olds using strings, sticks and hooks depending on perceptual properties of the task, such as the difference in color or texture between the tool and the toy, showing for instance that the task was more difficult if they had the same color and texture. Brown (1990) document the transfer abilities of 17 to 36 month olds in retrieval tasks with stick-like tools. In Rat-Fischer et al. (2012), the authors test the task of retrieving an out-of-reach toy with a rake-like tool with 14, 16, 18, 20 and 22-month-old infants, depending on physical properties of the task such as the spacial gap between the tool and the toy. They show that the youngest infants can spontaneously solve the task when the toy is inside or touches the tool, and 18 months can succeed at the task with a spatial gap. Also, starting from 18 months, infants can benefit from demonstrations of the task.

In these different studies, the task is set up as a problem to be solved, most often with one toy that should be retrieved, which can be done only through the use of a tool. In the second year of life, babies are not necessarily attracted by the “toy” more than the “tool” object that should be used as a tool to act on the toy, or than any other object in the testing room. Therefore, the toy is usually made attractive in many different ways: attractive in color and shape, attractive because the experimenter talks about the toy and points towards it, and/or attractive thanks to a training phase where similar toys are presented. Experimenters also remove any other unnecessary or distractor object from the room, both for a better control of the experiment and to focus the attention of the baby. The task starts when the objects are in place, and stops when the baby retrieves the toy, which is considered

as a success, or after some time when it is then considered as a failure. With such a setup, the task and its interpretation assume the baby is doing its best effort to retrieve the toy.

However, even with those precautions, the baby can still be distracted by the experimenter, the caregiver, any object left in the room, any object left to help with the experiment such as a camera, or any part of the setup that is not directly linked to the tool-use strategy or the toy. Also, when babies fail to retrieve the target object, they may switch their attention towards other objects. In many studies, a non-negligible proportion of baby experiments are removed from further analysis as babies were not motivated by the task, cried for not getting the toy, were fussy, were playing with something unrelated to the task, etc. Furthermore, when the baby is in appearance trying to solve the task, he might in fact be playing with one of its components, and not particularly willing to retrieve the toy. Indeed, babies are intrinsically motivated to explore their environment (see Background), spontaneously performing their own goal-directed actions (Von Hofsten, 2004).

We hypothesize that this success/failure/dropout framework for studying the development of sensorimotor skills hides or obscures the natural diversity of learning mechanisms in children, in particular intrinsically motivated exploration which is usually neglected in the design of those experimental protocols and may interfere in the interpretations of their results.

In this chapter, we provide a case-study to help evaluating this hypothesis. We reanalyze in more details a particular tool-use experiment that was performed by Lauriane Rat-Fischer, Megan Hamer (Dept Zoology, Univ. Oxford), Kim Plunkett (Dept Experimental Psychology, Univ. Oxford) and Alex Kacelnik (Dept Zoology, Univ. Oxford) (see Section 3.1). The original goal of this experiment was to test whether different types of pre-experience with objects influence 21-month-old infants performance at a sequential tool task. To this end, they designed a tool-use task where an attractive toy is placed inside a transparent tube, and the only way to retrieve the toy is to insert in sequence several wooden blocks in one side of the tube. There were initially 37 babies, first trained with one of four possible conditions: training with one long tool, training to get a reachable toy with the hand, training with a setup composed of sequential parts or no training, and then tested in three consecutive trials. Infants were quite successful at this task, with 24 infants that succeeded to reach the toy at least in one trial out of 32 analyzed. However, there was no significant difference in the success rate as a function of training conditions. This sequential tool-use task is expected to be more difficult than single tool-use tasks such as the one of Rat-Fischer et al. (2012) where about two third of 22 month olds solve a simpler rake task. Indeed, inserting several blocks in sequence in a same side of the tube should require more advanced planning skills than using a single tool to retrieve a toy. Still, the success rate was high and similar to the rake task, which seems counter-intuitive.

In order to understand infants' learning mechanisms in this task and in particular

the potential role of intrinsic motivations, we investigate in more details their behaviors and the relation between their behaviors and the progression of the task resolution. We first gather a body of anecdotal observations (Section 3.2), which describes several cases of behaviors that seem not to be driven by the goal of retrieving the toy to solve the task, but rather seem to be the result of an intrinsically motivated exploration with a diversity of potential alternative goals. Then, we analyze a corpus of videos from this experiment and annotate many behaviors in real-time, such as the actions and gaze of babies, events related to task, or behaviors of the experimenter and caregiver (Section 3.3). We focus on the whole testing period, even after the end of successful trials defined by the moment when the toy goes out of the tube.

The analysis of those annotations unveils a diversity of motivations and allows to characterize their associated behaviors to some extent (Section 3.3.2). The consideration of the intrinsically motivated spontaneous exploration of the environment by babies brings a possible explanation to the counter-intuitively high success rate in this sequential tool-use tasks (Section 3.4). Even if the environment is constrained by the attractiveness of the toy together with the experimenter further attracting the attention of the baby towards the toy, babies spend time exploring their own goals. One such alternative goal is to insert all objects into the tube, which often has the unpredicted consequence of pushing the toy out of the tube, by “chance”, or from a more operational point of view, by “intrinsically motivated exploration”. In those cases, the trial is labeled as a success, which could be the explanation of a high success rate, resulting from the interference of the own motivation of babies with the task goal as set by the experimenter.

3.1 A Sequential Tool-Use Experiment with 21-Month Olds

3.1.1 Ethics

This study has been ethically reviewed and received ethics clearance from the University of Oxford Central University Research Ethics Committee (Ref. MSD-IDREC-C2-2014-017).

3.1.2 Participants

A total of 37 healthy full-term infants aged 21 months participated in the study. Five infants were excluded from the final analyses due to fussiness (crying, $n=2$), lack of motivation to participate ($n=1$) or because they held something that was not part of the experimental set up in one of their hand during the testing phase (biscuit or toy, $n=2$). Those behaviors seem unrelated to the ability to solve the task. The remaining 32 infants (11 girls, 21 boys) were aged 619 to 656 days, with a mean age of 631 days.



Figure 3.1: Sequential tool-use task. A salient toy is placed inside a transparent tube open on both ends. Six wooden blocks are placed around the tube, at least two of which must be inserted on one side of the tube to push the toy out of the tube.

Infants were pseudo-randomly assigned to one of four pre-training groups: a Body training ($n=9$, 3 girls), an External training ($n=8$, 3 girls), a Functional training ($n=7$, 2 girls), and a “no training” group ($n=8$, 4 girls). Sex was counterbalanced as much as possible because some studies have found sex differences in tool-use performance (e.g. Chen et al. (2000)). Infants were recruited from a database of local families who had expressed interest in taking part in studies of infant development. Prior parental consent was granted before the infants took part in the study. All infants/parents whose face appears on the figures in this paper have further agreed to sign an optional image release agreement form.

3.1.3 Procedure

Upon arrival, infants were first given a short warm-up phase within the lab reception, during which the experimenter offered various toys and played with the infant to give them an opportunity to familiarize with both the experimenter and the environment. In parallel, the caregivers were given all information and consent forms, and were explained the overall procedure. The infant and caregiver were then transferred to a quiet testing room. To reduce potential stress from being tested by an unfamiliar experimenter in a novel environment, caregivers were present during the whole




Training Type	Body	External	Functional	No Training
Description	Infants are trained to solve the problem with part of their body (direct feedback)	Infants are trained to solve the problem with a pre-inserted object (indirect feedback)	Infants are trained with the function of the tool out of the context of solving a problem	Infants receive no training and are directly presented with the tool task straight after they have been given the laterality test (no feedback)
Training Task	 Use of the hand to push the toy out of the tube	 Push a pre-inserted object that will itself push the toy out of the tube	 Use of a set of objects to be moved together (one train carriage is used to push another)	

Figure 3.2: Training Conditions. One of 4 types of pre-experience are given before the sequential tool task.

experiment, but were asked to interfere the least possible, except for occasionally verbally encouraging their infant to interact with the objects and/or retrieve the toy from the apparatus, but without showing or telling them how to solve the task. The infant sat in the lap of their caregiver in front of a table, and an experimenter sat opposite, facing the infant. All training conditions and the test were prepared behind an opaque screen, to prevent infants from observing the experimenter. Additionally, the screen was positioned between the experimenter and the apparatus during each test trial, to prevent the experimenter from looking at the toy (but not at the infant) to avoid providing infants with directional cues.

3.1.4 Task

The task consisted in a transparent horizontal tube (26cm long perspex with a diameter of 6cm, the top is 20cm high) opened on both ends, and containing a small and colorful toy at the center. The tube was fixed on a wooden base (40cm in length, 20cm in width and 1cm in height). In the test phase, 6 small wooden blocks were scattered around the tube, and the toy could be pushed out of the tube by inserting at least two blocks from the same end of the tube (Figure 3.1).

3.1.5 Training Conditions

The training phase consisted of one out of four different training: Body, External, Functional or No training (see Fig. 3.2). In the Body training, there were no wooden blocks, the toy was placed at one end of the tube and infants were shown how to push the toy inside the tube with their own fingers. To facilitate the successful push

of the toy outside of the tube, a small wedge was fixed under one side of the wooden base to raise it slightly aiding the infant in pushing the car toward the other end. The aim of the training was to give infants a direct feedback on how to act on the toy to push it out of the tube. In the External training, a 20-cm-long wooden block was pre-inserted into the Tube, one end sticking out for about 3 to 5cm. Infants were shown how to push the toy with the pre-inserted block by pushing directly on the block. The aim of the training was to give infants an indirect feedback of how to act on the toy to push it out of the tube. In the Functional training, the apparatus was replaced by 3 disconnected wooden train carriages, and infants were shown how to push the carriages by moving only one of the carriages. The aim of this training was to familiarize infants with the sequentiality of moving two objects with a third one, thus familiarizing them with the sequential tool functionality. Finally, in the No training condition, infants were not familiarized with either the apparatus or the tool's functionality. To control for social and manipulative exposure, infants in this condition were presented with a short additional test for laterality (Fagard and Marks, 2000).

During the training phase the experimenter demonstrated to the infant how to successfully reach the toy. To reach the success criteria, infants had to succeed the training condition at least 3 times in a row without demonstration. All infants successfully reached the success criteria within 3 trials/demonstration events, and were then presented with 3 test trials. In the rare cases ($n=6$) where infants did not reach the toy at least once during the first two trials, they were not given a third trial, to prevent frustration.

3.1.6 Performance Analysis

There were 64 successful trials out of 93 trials in total. If we consider each of the three trials separately, there were 22 successes out of 32 first trials, 20 successes out of 32 second trials, and 20 successes out of 27 third trials.

Number of infants who succeeded at least once. We found no difference between training groups in the number of infants who successfully reached the toy at least once ($\chi^2(3, 32) = 1.76, p = .62$; Body 6/9, External 7/8, Function 6/7, None 7/8).

Mean Number of trials to reach 1st success. Most infants were able to reach the toy in their first trial ($n=21/32$), and for the 21 successful infants, there was no significant difference between training groups in the number of trials needed to reach the toy (Kruskal Wallis, $\chi^2(3, 32) = 1, p = .8$; mean Body = 1.17, External = 1.14, Functional = 1, None = 1.14).

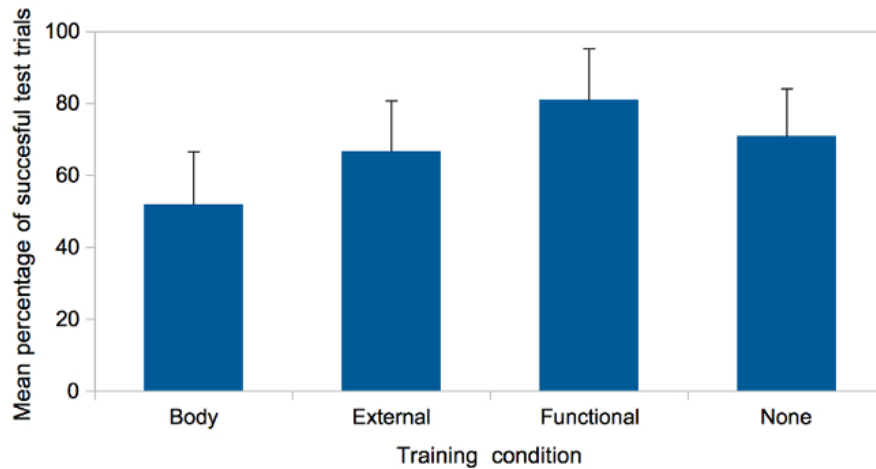


Figure 3.3: Mean percentage of successful test trials as a function of training condition.

Mean Proportion of successful trials. Despite an apparent tendency to be less often successful after the Body training and more often successful after Functional training, we failed to find a significant difference between training groups in the mean percentage of successful test trials (Kruskal-Wallis, $\chi^2(3, 32) = 2.36$, $p = .5$, Figure 3.3).

Time to first success. The hypothesis is that in the Body and None training conditions, infants need more time to reach their first success than in the External and Functional training conditions, because they are not familiarized with the behavior of pushing an object that itself pushes another one. Despite a tendency suggesting that infants not exposed to any kind of training need more time to reach their first success, there was no significant difference between training groups (Kruskal-Wallis, $\chi^2(3, 32) = 6.25$, $p = .10$, mean Body = 56s, External = 60s, Functional = 46s, None = 104s, Figure 3.4).

Time to succeed at each trial. A GLMM analysis on the time needed to succeed as a function of trial number (1, 2 or 3) and training condition (body, external, functional or none) revealed a significant interaction between the two factors: infants who were not exposed to any kind of training needed more time on their first trial to reach success than infants exposed to the external and functional training (znone-external = 65.92, $p < 0.05$; znone-functional = 61.99, $p < 0.05$, Figure 3.5), but not the body training (znone-body = 54.78, $p = 0.12$). There was a significant difference between trial 1 and the two subsequent trials for the condition without training only (ztrial1-trial2 = 73.81, $p < 0.01$; ztrial1-trial3 = 62.57, $p < 0.01$). This result suggest that any kind of training (involving or not the apparatus) may facilitate infants' first success, whereas infants not exposed to any form of training may need more time to

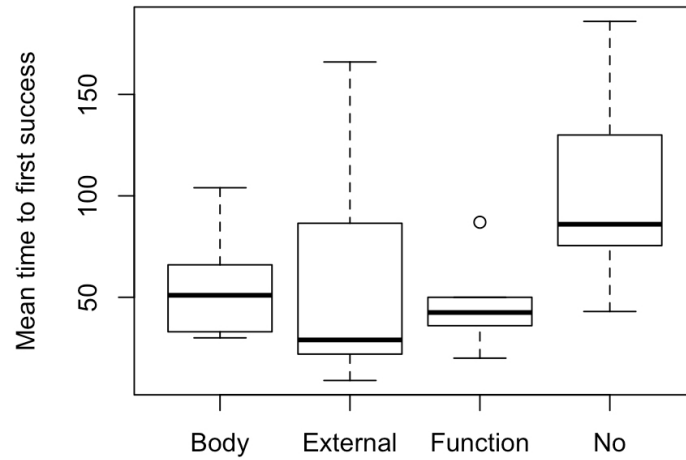


Figure 3.4: Time (in seconds) to reach the toy for the first time during the test trials as a function of training condition.

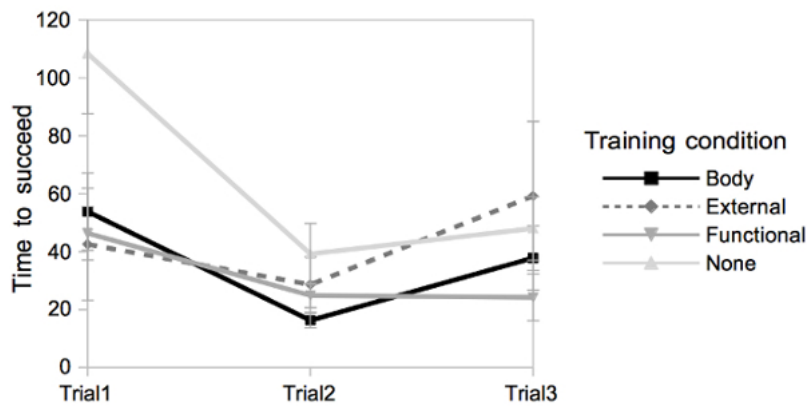


Figure 3.5: Time (in seconds) to reach the toy as a function of training condition and trial number.

explore the apparatus before being able to reach the toy successfully.

3.2 Anecdotal Observations of Intrinsically Motivated Behaviors

In this section, we gather a small body of anecdotal children observations from this tool-use experiment, reporting behaviors that seem not to be driven by the only goal of retrieving the toy to solve the task, but rather seem to be the result of an intrinsically motivated exploration with a diversity of potential alternative goals and strategies.

The first observation (C2-1) describes a discovery of the use of the block as a tool to move the toy inside the tube.

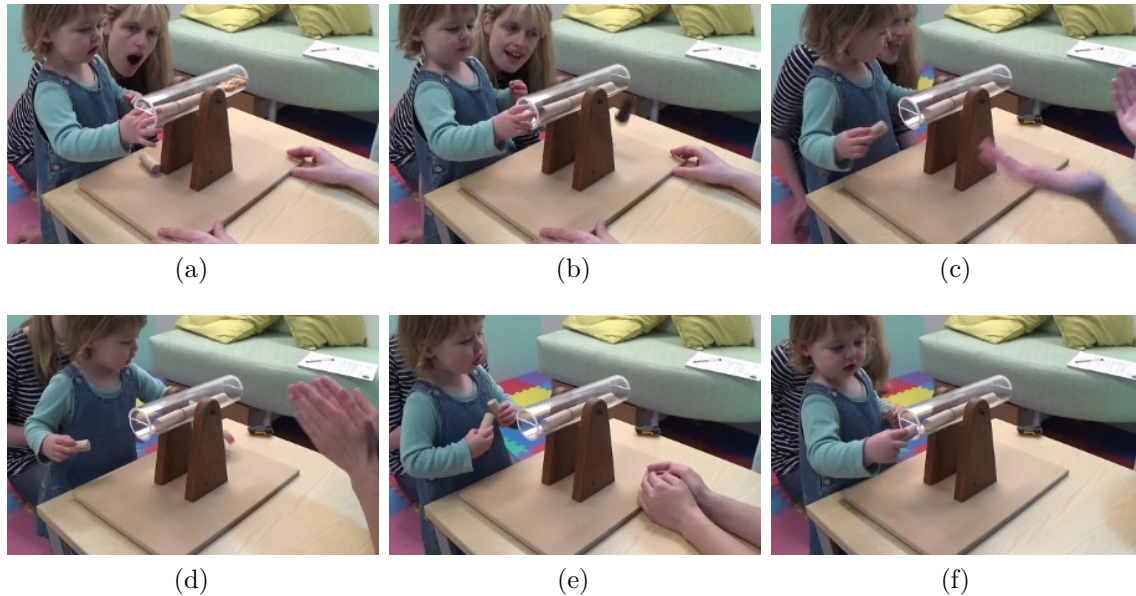


Figure 3.6: Baby H2 saying “Oh, no!” after the toy falls out of the tube.

Observation C2-1. Baby C2 was trained with the long version of the tool (External training condition). In trial 1, she inserts one short tool, but like in training, she does not drop the tool, but get it out instead. Once, she points at the cardboard box containing the long version of the tool, among other toys, while vocalizing and looking at the experimenter, as if she was asking for the longer version of the tool since the current tool was too short to push the toy out. She also often looks at the experimenter and holds a tool out to the experimenter in trial 1 and 2. In trial 2, a small smile can be observed when the toy is pushed a little bit after dropping a block in the tube and pushing it further inside with the hand. A moment later, she inserts and pushes a second block inside the tube which largely moves the toy, at which point a big smile is observed. The insertion of a third and fourth block makes the toy fall out of the tube. In trial 3, the baby directly inserts three tools and quickly succeeds to retrieve the toy. At the starts of trial 4, one can see baby C2 nodding when the experimenter pushes the apparatus towards her.

As shown by her failed previous attempts and by the smile appearing after the movement, the baby seems to be surprised by the first movement of the toy pushed by the tool, which happen when exploring the interaction of the wooden blocks and the apparatus.

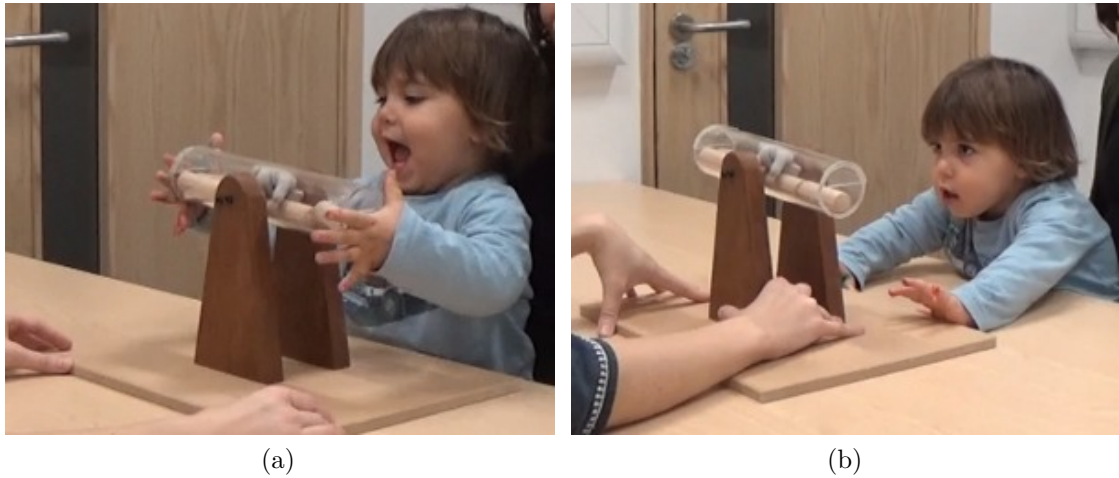


Figure 3.7: Baby N1 (a) looking at the car and saying “Yay!”, (b) just after, he pushes the apparatus towards the experimenter.

The two following observations (H2-1 and N1-1) show clear examples of a discrepancy between the goal of the task, and the goal of the baby, betrayed by the vocalizations of the baby.

Observation H2-1. Baby H2, in her first trial, inserts one block into the tube on the right side, while saying “There!”. Then, she inserts a second block, which pushes the first one and the car, while saying “Choo-Choo” multiple times, and displays the same behavior with a third block (Figure 3.6a). The insertion of a fourth block makes the toy fall out (Fig. 3.6b). The experimenter and caregiver immediately say “Yay” (Fig. 3.6c), and the experimenter applauds. The experimenter then says “Well done!” while the baby briefly looks at the toy and grabs another block near the toy (Fig. 3.6d). The baby then says “Oh, no!” (Fig. 3.6e, less than 3s after the fall) while the caregiver is still saying “Yay” and started applauding. After, the baby continues inserting blocks as before (Fig. 3.6f), with no further interest in the toy.

Observation N1-1. Baby N1 succeeds the first trial, seemingly by chance while trying to insert all blocks into the tube, and fails the second trial, as inserting two blocks on each side did not move the toy out. In the third trial, the baby also inserts two blocks on each side, pushes on both sides and says “Yay!” while looking at the toy (Figure 3.7a). Just after, he pushes the apparatus towards the experimenter, as if the task was finished (Figure 3.7b).

In the three following observations, babies seem to be pursuing their own alternative goals, such as inserting blocks into the tube or drawing with a block.

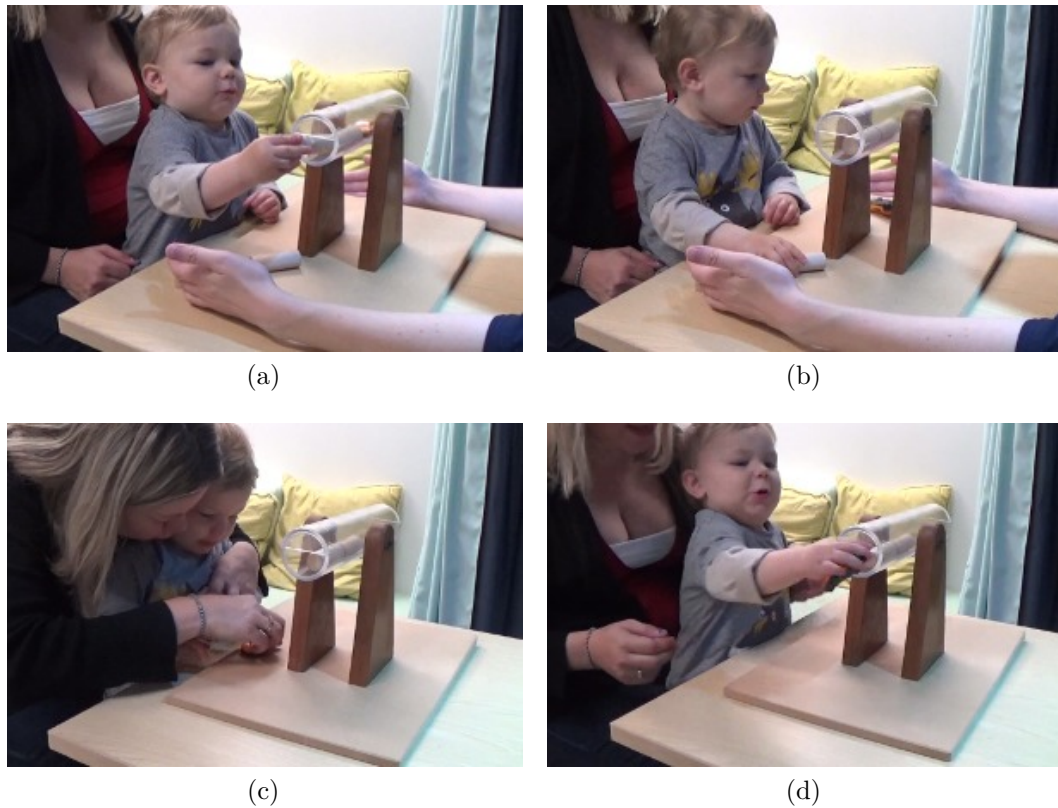


Figure 3.8: Baby A3 replacing the toy in the tube.

Observation A3-1. In trial 1, baby A3 pushes the toy out of the tube by inserting a third block. He does not immediately see that the toy is out, but the caregiver and the experimenter says “Yay!”, the baby then looks at the toy out of the tube, the caregiver applauds, the baby takes the toy with its hand. The baby immediately inserts the toy back into the tube from where it got out.

Observation A3-2. In trial 3, after inserting a second block into the tube (Figure 3.8a), the toy falls out of the tube while the baby looks at and grabs a third toy. This fall produces noise from the toy and “Yay!” from the caregiver and experimenter. The baby then looks at the toy (Figure 3.8b) and grabs it, while the caregiver applauds and the experimenter says “Well done!”. The caregiver then takes the car toy from the hands of the baby and demonstrate one of its functions: rolling like a car, while saying “Look! Broom broom!” (Figure 3.8c). The baby then takes the car from the hands of its caregiver and immediately puts it back into the tube (Figure 3.8d), fails, and tries again.

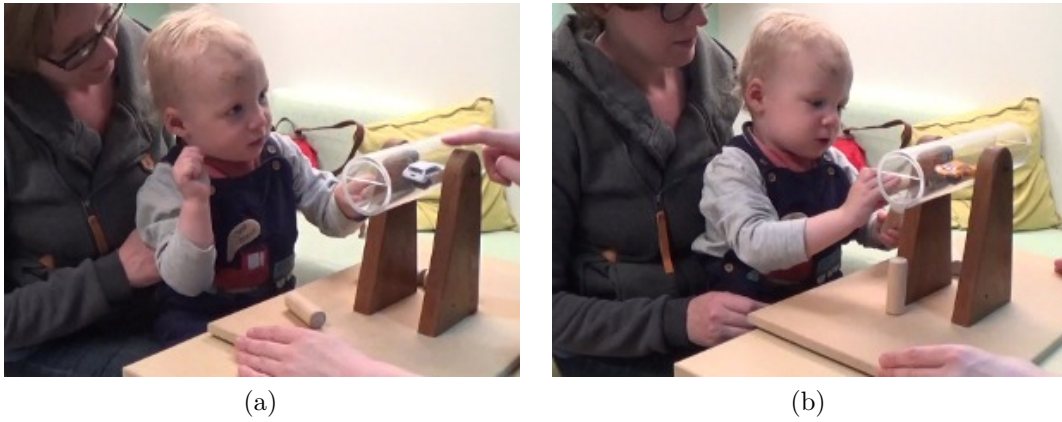


Figure 3.9: Baby A2 “drawing” with the wooden block on the apparatus.

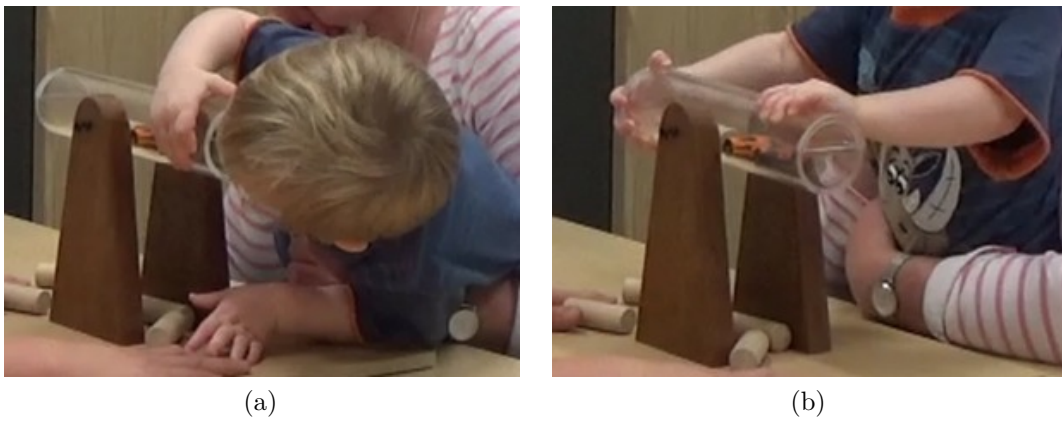


Figure 3.10: Baby H3 (a) looking at the car and saying “Car, hello!”, and (b) trying to rotate the tube.

Observation A2-1. Baby A2 is more interested by “drawing” with a wooden block on the apparatus, while saying “Chalk!” many times, although this do not actually draw, and despite the experimenter trying to attract his attention towards the toy. He displays this behavior in the most part of the three successive trials, and inserts a wooden block in the tube only once in the first trial, before putting it out and continuing to “draw”, thus resulting in three failed trials. Figure 3.9 shows two examples of this behavior, in the first trial (a) and in the second trial (b).

Lastly, we report observations of a baby that wants to retrieve the toy but that spontaneously tries new strategies, other than the only functioning strategy using blocks as tools.

Observation H3-1. After successfully pushing the toy out of the tube in the first two trials, baby H3 starts the third trial by inserting a block in the tube, and tries to get the toy with the hand on the other side, without success. H3 then leans forward to look directly at the car from the left hole of the tube, and says: “Car, hello!”.

Observation H3-2. Shortly after Observation H3-1, baby H3 straightens above the tube and tries to rotate the tube with both arms and all of his strength, alternating the rotation in one direction and the other, supposedly so that the car falls by gravity. However, the tube cannot rotate enough for this strategy to work and the toy only moves a little bit (Figure 3.10b).

3.3 Fine-grained Analysis of Behaviors and Events

In order to understand the exploratory behaviors of infants and their motivations in the tool-use task, we designed a set of events and behaviors that we annotated with a fine-grained time scale from videos of the tool-use experiment. A complete list of annotated behaviors and their description can be found in Table A.1 of Appendix A.

3.3.1 Methods

The general approach we used to define those behaviors was to take into account all the possible exploratory behaviors of infants. Therefore we did not limit the analysis to their manual interaction with the tool-use apparatus, but we also annotated their actions not directly related to the realization of the task but that could be related to their understanding of the task or to their state of motivation towards solving the task or towards other objects. This includes the direction of their gaze, their possible vocalizations and other expressions or social behaviors.

Infants’ motivations and behaviors may also be influenced by other events in the tool-use setup and more generally in the environment. We thus paid particular attention to the vocal and manual inputs from the experimenter, that meant to attract the attention of the infant towards the toy or towards the goal of retrieving the toy, as well as to the encouragements from the caregiver. We also took into account the changes in position of the toy inside the tube as we assumed it could have an influence beyond the fact that the toy is inside or outside the tube.

The microgenetic approach for the study of behavioral changes recommends to observe behaviors on a trial-by-trial basis over a large period of time (Chen et al., 2000). Although we study behaviors and motivations in only one tool-use session here, we measure those behaviors with an even higher density, with a precision of less than the second for the fastest actions such as gaze direction.

In the context of the study of babies' intrinsic motivations in tool-use tasks, we consider that the time boundaries of trials as defined by the experimenter are not appropriate boundaries for the observation of intrinsically motivated exploration. We thus continue to observe and annotate behaviors in between trials, from the point where the experimenter decided that a trial is a failure or from the point where the baby succeeded to get the toy out of the tube, to the point where the apparatus is removed from its reach. In the case where the baby succeeded to put the toy out, we think that the actions of the baby in this period in between trials convey important information about the motivation to get or play with the toy.

The previously described behaviors had four possible subjects: the experimenter, caregiver, environment and baby. For the experimenter, we coded its actions with the apparatus, its social feedback to attract the attention of the baby, and the starting and ending of trials and phases. For the caregiver, we annotated social feedback such as attracting the attention of the baby towards the task, spoiling the solution of the tool-use problem in few cases, expressing encouragements or surprise. The environment category deals with the changes of position of the toy that is placed by the experimenter in the middle of the tube at the beginning of each trial, describing the position of the toy with a 7-point scale going from one end to the other end of the tube in addition to the state where the toy is out of the tube.

Additionally, we coded the goals and strategies to reach those goals that the baby could be having at any point in the tool-use experiment. Since these annotations are subject to the interpretations of the observer/coder, they are subjective, as opposed to the previously described behaviors. We did not specify particular properties for a behavioral sequence to be compatible with a given goal and associated strategy, the annotations were rather made from observer's intuition and should be interpreted accordingly. The annotated goals were the following: to retrieve the toy, to retrieve a tool, to insert all wooden blocks in the tube, to use a wooden block as a tool, to place the toy back in the tube after succeeding the task, to draw on the apparatus, and to make noise. The strategies were to use directly the hand or to use a tool to try to retrieve the toy, to seek for the help of the caregiver or experimenter, to exchange the currently held tool with another one or to switch the hand holding the tool. Those goals and strategies are labeled with the "Observer" subject, which also include the success or failure of the trial and an interpretation of the success, from success by chance to fully intentional success.

The set of behaviors and events has been incrementally defined and refined through partial coding of data, and we recoded the experiments with all babies with this final set as detailed in Table A.1. The videos were annotated with the BORIS software (Behavioral Observation Research Interactive Software) (Friard and Gamba, 2016).

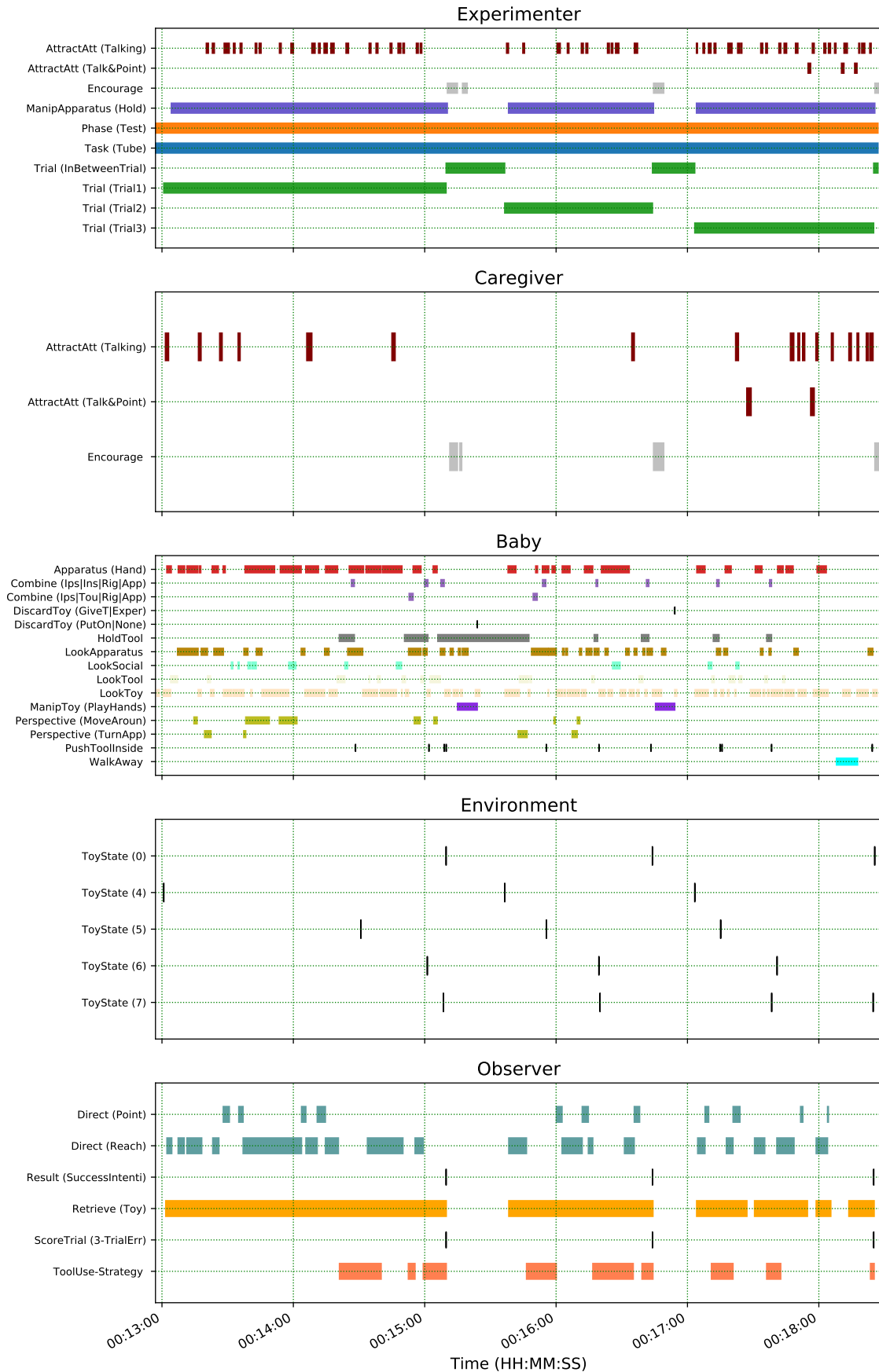


Figure 3.11: Ethogram of experiment with Baby C1.

3.3.2 Results

The annotations of behaviors and events result in one “ethogram” per baby. Figure 3.11 shows one example of an ethogram of the test phase of the experiment with one baby. See Appendix A for ethograms of several other babies. In those figures, we plot on the Y axis the different behaviors and their properties (also called behavior “modifiers”), while the X axis is the time from the start of the video. We analyze the whole test phase, containing 3 to 4 trials plus the periods between trials and just after the end of the last trial until the apparatus is removed. Two infants were excluded from the gaze analysis (but were included in all other analysis) because the point of view of the camera did not allow to reliably annotate gaze direction.

In the following, we study several aspects of experiment’s progress that we hypothesize to be related to the intrinsic motivations of babies to explore the apparatus. We first look at the variability of babies’ behaviors as an indicator of their active learning in this task. We also focus in particular on the moments when the toy falls out of the tube as we assume that if the babies really want to get the toy, then they will grab and play with it once it becomes within reach. As the behaviors with the toy at that moment may be linked to the goal they had during the trial, we then study their fine-grained actions and looks, in general and depending on the behaviors with the toy once it is within reach. Finally, the experimenter talking and pointing can interfere with the intrinsic motivations in the choice of goals and strategies, so we study the reaction of babies to the experimenter attracting their attention.

Behavioral Variability

We annotated different behaviors concerning the babies’ hands, the tools and the apparatus. Most babies displayed several behaviors in the same trial. Here we provide a measure of the behavioral variability of a baby as the mean over trials of the number of different behaviors that were displayed in a same trial. We include the following list of babies’ behaviors in this measure: Apparatus, Combine, PlayTool, HoldTool, PushToolInside, Perspective, SwapTool-A, SwitchHand, DiscardTool. In order to discriminate all coded behaviors, each different set of modifiers (specifying variations of behaviors) for those behaviors counts as a different behaviors. For instance, using the left hand to insert a block on the left side of the tube, using the left hand to insert on the right side, using the right hand to insert on the right side, are counted as a different behaviors in this measure.

The mean behavioral variability over 32 babies is 5.1 different behaviors per trial (min: 2.0, max: 9.5, std: 1.5). The variability is similar in successful and failed trials: with a mean of 5.2 different behaviors per successful trial (26 babies with at least one successful trial), and 5.5 different behaviors per failed trial (18 babies with at least one failed trial). The variability is also similar in trials happening after or before a successful trial: with a mean of 4.9 different behaviors per trial after a successful trial,

and a mean of 5.6 behaviors per trial otherwise. Figure 3.12 shows an histogram of the number of different behaviors per trial.

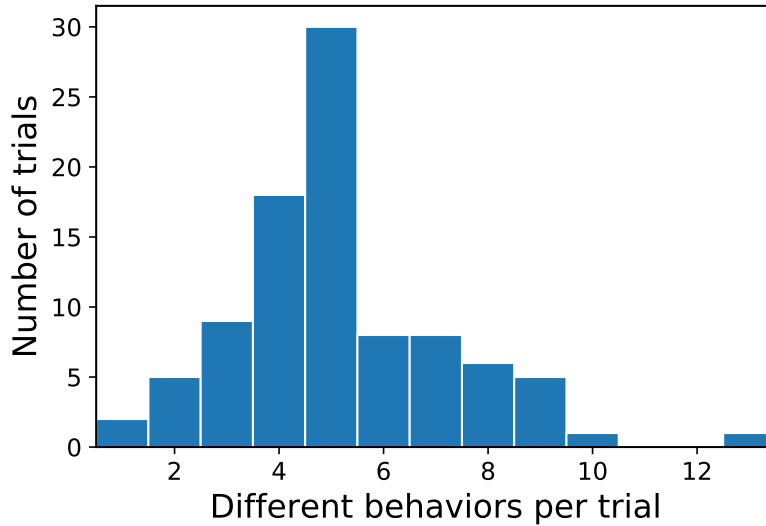


Figure 3.12: Histogram of the behavioral variability in trials.

When the Toy Goes Out

In some of the following analysis, we focus on the moments when the toy got out of the tube in successful trials. Here we define a categorization of those trials depending on the behavior of the baby with and without the toy once the toy is out. There were 64 such trials in total out of 93 trials for the 32 babies. We grouped those trials into several categories based on the behavior of the baby once the toy is out. In the PlaceToyBack category, the baby put the toy back into the tube within 15 seconds after the toy got out (6 trials, 5 babies). In the ManipulateToy category, the baby touched the toy within 5 seconds and did not put it back into the tube (33 trials, 16 babies). In the CombineTool category, the baby combined a tool with the apparatus within 5 seconds and did not put the toy back into the tube or manipulate the toy (16 trials, 10 babies). If the behavior did not fall into one of the previous categories, it is labeled as Other (9 trials, 6 babies).

Hand/Tool Actions

Infants spent 21% of the experiment time exploring the apparatus with the hand, and 13% of the time combining a tool with the apparatus or other tools. There was no significant difference in the proportion of time spent exploring the apparatus with the hand in the 3 trials (Kruskal-Wallis, $\chi^2 = 1.64$, $p = .44$, mean Trial 1: 29%, mean Trial 2: 28%, mean Trial 3: 21%), nor combining a tool with the apparatus or other

tools in the 3 trials (Kruskal-Wallis, $\chi^2 = 2.26$, $p = .32$, mean Trial 1: 22%, mean Trial 2: 17%, mean Trial 3: 20%).

We can look in particular at the moments when the toy moves inside the tube during trials (changes in ToyState) and compare a 5-second period after versus before the toy moves. Infants spend less time combining a tool with the apparatus or other tools after the toy moves than before: (Mann-Whitney, $U = 159$, $p < .0001$, mean after: 18%, mean before: 32%). There was a slight but non-significant increase in the time spent exploring the apparatus with the hand after the toy moved (Mann-Whitney, $U = 290$, $p = .19$, mean after: 22%, mean before: 18%).

However, babies spend more time exploring the apparatus with the hand after the toy moves in the trials of category ManipulateToy, than in the CombineTool trials (Kruskal-Wallis, $\chi^2 = 4.8$, $p = .028$, mean ManipulateToy: 26.5%, mean CombineTool: 3.6%). Also, they spend less time combining the tool with the apparatus or another tool after the toy moves in the ManipulateToy trials than in the CombineTool trials (Kruskal-Wallis, $\chi^2 = 4.7$, $p = .031$, mean ManipulateToy: 17.7%, mean CombineTool: 31.8%).

Finally, there was no significant difference in the time spent manipulating the apparatus with the hand depending on the goal or strategy assigned by the observer (including only the ones that use wooden blocks: InsertAll, ToolUse-Goal and ToolUse-Strategy) (Kruskal-Wallis, $\chi^2 = 2.21$, $p = .33$, mean InsertAll: 7%, mean ToolUse-Goal: 4%, mean ToolUse-Strategy: 16%), nor in the time spent combining the tool with the apparatus or another tool (Kruskal-Wallis, $\chi^2 = 0.8$, $p = .668$, mean InsertAll: 33%, mean ToolUse-Goal: 45%, mean ToolUse-Strategy: 34%).

Gaze

In general, the mean percentage of time looking at the toy in the experiment is 28.2%, and the mean percentage of time looking at a tool is 17.3%. This pattern does not significantly depend on the success status of trials: the mean percentage of time looking at the toy is 29.9% in successful trials and 32.3% in failed trials (Mann-Whitney, $U = 191$, $p = .266$), while the percentage of time looking at the tool is respectively 25.5% and 22.9% (Mann-Whitney, $U = 183$, $p = .204$).

When the toy falls out of the tube, babies' looking pattern changes. They look more at the toy in the 3 seconds after the toy goes out of the tube (67.1%) than in the 3 seconds before (26.7%) (Mann-Whitney, $U = 65$, $p < .001$). Also, babies look less at the tool in the 3 seconds after (13.2%) than before (23.6%) (Mann-Whitney, $U = 207$, $p = .048$).

We can consider their looking pattern after the toy moves depending on their behavior after the toy goes out of the tube in successful trials, whether they manipulate the toy or combine a tool with the apparatus or another tool. They look more at the toy after it moves in category ManipulateToy (49.2%, 15 babies) than in category CombineTool (17.2%, 7 babies) (Kruskal-Wallis, $\chi^2 = 6.806$, $p = .009$). Also, they

look less at the tool after the toy moves in category ManipulateToy (15.8%) than in category CombineTool (29.9%) (Kruskal-Wallis, $\chi^2 = 3.123$, $p = .077$).

We found no significant difference in the percentage of time looking at the toy depending on the observed goal and strategies that use tools (ToolUse-Strategy 27.5% for 21 babies, ToolUse-Goal 15.3% for 6 babies, and InsertAll 20.6% for 19 babies) (Kruskal-Wallis, $\chi^2 = 3.456$, $p = .177$). There may be a difference in the percentage of time looking at the tool depending on the observed goal and strategies that use tools (ToolUse-Strategy 32.1%, ToolUse-Goal 51.6%, InsertAll 32.9%) (Kruskal-Wallis, $\chi^2 = 5.610$, $p = .060$). A post-hoc pairwise test indicates a significant difference between ToolUse-Strategy (32.1%) and ToolUse-Goal (51.6%) (Kruskal-Wallis, $\chi^2 = 4.049$, $p = .044$).

We also provide a measure of the frequency of switches between looks at the toy and looks at the tool. The overall switch frequency is 0.118 switch per second. The switch frequency depends on the success status of trials, with more frequent switches in success trials (0.157 switch per second) than in failed trials (0.120 switch per second) (Mann-Whitney, $U = 146$, $p = .038$).

Attracting Infant's Attention

In this experiment, the caregiver is asked to encourage the baby but not to help with solving the task. The experimenter also encourages the baby, and furthermore regularly attracts her attention towards the task, either by talking, by pointing towards the toy, or both. Here, we measure the quantity of those behaviors and their influence on the actions of the baby.

During trials, the experimenter talks without pointing on average 10.8% of the time, points without talking 0.07% of the time, talks and points at the same time in 3.3% of the time, and encourages the baby 0.18% of the trial time. The caregiver talks 2.6% of the time, and encourages the baby 0.88% of the time.

The experimenter attracts the attention of the baby towards the toy by talking more in failure trials (12.6%) than in success trials (8.3%) (Mann-Whitney, $U = 148$, $p = .020$).

When the experimenter talks, this does not change the average time spent by babies looking at the toy or the tool in the 3 seconds after talking versus in the 3 seconds before talking (toy: 34.4% vs 34.1%, tool: 18.9% vs 17.1%). However, when the experimenter talks and points at the same time, the time looking at the toy increases from 11.1% to 43.2% (Mann-Whitney, $U = 89$, $p < .001$), and the time looking at the tool decreases from 37.5% down to 16.0% (Mann-Whitney, $U = 159$, $p < .001$). If we consider the behavior of the experimenter depending on the previous looking behaviors of the babies, we found that the experimenter talks and points (at the same time) more in the 3 seconds after the babies look at the tool than in the 3 seconds after the babies look at the toy (Mann-Whitney, $U = 219$, $p < .001$): the experimenter is talking and pointing 4.26% of the time versus 1.05%.

The actions towards the apparatus or the tools may also be influenced by the talking and pointing behaviors of the experimenter. The percentage of time exploring the apparatus with the hand does not significantly change after the experimenter talks compared to just before (31.8% vs 35.2%) (Mann-Whitney, $U = 459$, $p = .240$), but the time spent combining a tool with the apparatus or other tools increases (13.5% vs 6.3%) (Mann-Whitney, $U = 288$, $p = .001$). However, the time exploring the apparatus with the hand does increase after the experimenter talks and points towards the toy at the same time (22.6% after vs 7.2% before) (Mann-Whitney, $U = 253$, $p = .009$) and the time spent combining a tool with the apparatus or other tools also increases (14.3% vs 8.4%) (Mann-Whitney, $U = 272$, $p = .020$).

3.4 Discussion

In this tool-use experiment, most infants succeeded in the first trial (21 out of 32), and 24 infants succeeded in at least in one trial. There was no significant difference in the success rate as a function of training conditions, or as a function of trial number. This sequential tool-use task was expected to be more difficult than single tool-use tasks such as the one of Rat-Fischer et al. (2012). In that experiment, the authors tested the task of retrieving an out-of-reach toy with a rake-like tool with 14, 16, 18, 20 and 22-month-old infants. They studied infants' tool-use abilities depending on the physical properties of the task, such as the spatial gap between the tool and the toy. With a small spatial gap, about 80% of 22-month olds solved the rake task, while with a large spatial gap, about one third succeeded.

The sequential tool-use task described in Section 3.1 requires the successive and consistent use of at least 2 tools to push the toy out of the tube. We therefore expected the sequential task to be more difficult than the rake tool task even with a large spatial gap, as it should require more advanced planning skills than using a single tool. Still, the success rate in the sequential task at 21 months was higher than the success rate in the rake task with a large spatial gap at 22 months, which seems counter-intuitive.

To understand the motivations of infants and their exploration and learning mechanisms in the sequential tool task, we investigated in detail their behaviors, goals and strategies. Overall, the most striking observation was the richness of the displayed behaviors and of the apparent goals and strategies of babies, despite the fact that only one type of behaviors allowed to solve the task (inserting at least 2 blocks on the left or right side).

Many anecdotal observations report the possibility of the babies pursuing alternative goals: other than the goal of retrieving the toy. Observations H2-1 and N1-1 are archetypal instances of such alternative goals, with a baby that is unhappy with the outcome of its actions while it actually is the one expected by the experimenter, and another baby that expresses satisfaction for its own behavior while it is completely

different from the target goal. In Observation H2-1, the baby seems to be considering the toy and several blocks inserted in the tube as a train, so that when the “head” of the train, which is the toy, falls out of the tube, she says “Oh, no!” while the experimenter and caregiver say “Yay!”, and continues inserting blocks into the tube. In Observation N1-1, the baby inserts two blocks on each side of the tube, compressing the elephant toy, says “Yay!”, and then pushes the whole apparatus towards the experimenter as if the game was over. His goal may have been to insert many blocks into the tube, or to crush the toy, however it does not seem to be the toy retrieval. Other potential goals have been reported, such as drawing on the apparatus with a block (A2-1), or placing the toy back into the tube once it is out instead of playing with it (A3-1 and A3-2). Also, when it seems that the current goal of the baby is to get the toy out of the tube, many strategies have been observed, other than inserting blocks to push the toy out. In Observation H3-1, the baby may be trying to talk the car out (“Car, hello!”), while in Observation H3-2, he tries to rotate the tube, which could have pushed the toy out through gravity, but did not, as this is not allowed by the apparatus. Other babies have been observed trying to trigger the caregiver’s or experimenter’s help, by saying “Please!”, “Help!”, or “Out!”, and another baby did obtain the toy directly from the experimenter after crying in several trials, but we don’t know if it was on purpose.

In light of these observations, we designed a set of events and behaviors that we annotated with a fine-grained time scale in the videos of the tool-use experiment. It includes the gaze and actions of the babies, the reactions of the caregiver and experimenter, the position of the toy inside the tube, and a subjective evaluation of the goals and strategies displayed by the baby.

We first measured the behavioral variability of infants as the number of different behaviors displayed per trial, resulting in an average of 5.1 different behaviors per trial. Most infants were actively exploring the setup, even if they already discovered the tool-use strategy to retrieve the toy.

If a baby wants to get and play with the toy, we expect him to actually play with the toy when it falls out of the tube. Therefore, we focused in particular on the period between the moment when the toy goes out, which marks the end of the trial from the point of view of the experimenter, and the moment when the apparatus and the toy are actually moved away from the baby. We observed several typical behaviors in this period just after the toy was pushed out: playing with the toy, placing the toy back in the tube, and combining a tool with the apparatus. In about half of the successful trials (33 out of 64), babies touch or play with the toy, while in the other half, they do not. In that case, they either place the toy back into the tube (6 trials), or combine a tool with the apparatus (16 trials), or otherwise (9 trials uncategorized). The motivations of infants in this task thus seem more diverse than initially expected.

We further analyzed the behaviors of babies during trials as a function of the category of their behavior after the toy got out of tube. We showed that the behaviors with the hand and the tools during the trials are consistent with the behavior after

the toy is out: babies that manipulate the toy after the toy is out use more the hand, less the tool, look more at the toy and less at the tool when the toy moves inside the tube, compared to babies that combine a tool with the apparatus after the toy is out. Those behaviors during the trial could thus be a correlate of the current goal of the babies, whether it is to retrieve and play with the toy or to just insert objects into the tube. One significant correlation between those behaviors and our subjective annotations of babies' goals and strategies is that babies are looking more at the tool when they were assigned a tool-use goal (52% of that time looking at the tool) than when assigned a tool-use strategy for the toy retrieval goal (32%).

The alternative goal of inserting objects into the tube seemed to be quite common among infants, as reported by our anecdotal observations and analysis of behaviors. For instance, Observation H2-1 reports the apparent goal of inserting blocks into the tube, with the baby saying "Oh, no!" when the toy falls out of the tube. Also, in the 10 trials where there was only one goal or strategy annotated, this was the InsertAll goal, with 9 of those 10 trials being successful. Observation C2-1 also shows that exploratory behaviors such as inserting objects can lead to the discovery of a strategy to solve the problem imposed by the task, and to the learning and reuse of this strategy in subsequent trials. In the first trial, baby C2 inserts one tool into the tube but fails to move the toy, while in the second trial, she discovers that she can move the toy by pushing a block into the tube, smiles and repeats this action by inserting three other blocks which makes the toy fall out of the tube. In the following trials, she directly uses the same strategy and quickly solves the problem. An intrinsic motivation pushing the baby to explore the apparatus could thus be the reason for the high success rate observed in this sequential tool-use experiment compared to other single tool experiments.

Overall, we observed a diverse set of alternative goals and strategies, some leading to the discovery of the appropriate strategy for solving the task. We reported that babies play with the toy in only half of the successful trials, while some of their behaviors during trials correlate with their actions after the toy is out. Those results confirm that the motivations of babies during and at the end of trials are diverse and do not necessarily coincide with the toy retrieval goal expected by the experimenter, despite the fact that the experimenter is attracting the attention of the baby towards that target goal. This diversity of behaviors, goals and strategies, given that only one goal is made salient, suggests that intrinsically motivated exploration plays a key operational role in driving goal selection, strategic planning and learning in such tool-use experiments. Our analysis of the original tool-use study shows that it is interesting to investigate how the use of tools can be discovered: in most cases it seems to be a collateral effect of an exploration with diverse goals more than the result of planning and reasoning about the particular goal of retrieving the toy.

In the tool-use study we analyzed, the experimenter attracted the attention of the baby towards the toy, by talking, pointing, or talking and pointing at the same time, for instance saying "Can you get the toy out? How can you get it out?". The

time spent attracting the attention of the baby was about 14% of trials' time, which is quite a lot. When the experimenter talked, we did not see babies changing their gaze patterns, however when the experimenter pointed towards the toy, babies looked much more at the toy and much less at the tool following this pointing. Also, talking made babies increase the time spent combining a tool with the apparatus, and talking and pointing at the same time made babies spend more time exploring the apparatus with the hand and combining a tool with the apparatus compared with before the talking. Talking and pointing had thus some effect in attracting the attention of the baby towards the toy, and was often used as a reaction to the baby getting bored by the task. Congratulating the baby when the toy goes out may also play a role in focusing the baby on this goal in the following trials. However, heavily attracting the attention towards the toy may inhibit the spontaneous exploration of tools and other elements of the setup and more generally of the environment. Given the limited attentional and processing abilities of 21-month olds, it could also interfere with the autonomous selection and remembering of goals, and learning of the solutions to diverse goals. The question of the interplay between intrinsic goals and strategies and extrinsic feedback and guidance deserves future work.

However, as pointed by Von Hofsten (2004), the role of motivations is often neglected in research on sensorimotor development. Many experiments studying infants' particular tool-use skills and strategies depending on their age do not acknowledge the potential role of intrinsic motivations in the selection of goals and strategies to explore. For instance, Koslowski and Bruner (1972) study a task where a lever has to be rotated to retrieve a toy. The toy is attached to the side of the lever that is not reachable at the beginning of the experiment, but can be made reachable by rotating the lever's side that is reachable at the beginning. One difficulty of the task is that the accessible lever side has to be pushed away so that the toy comes closer. They tested children from 12 to 24 months old, and observed several behaviors: direct, where children try to get the toy directly with the hand, oscillation of the lever, partial rotation, play with the rotation ignoring the toy even if it got within reach, and rotate the lever and grasp the toy. From 12 to 24 months, they observe an increase of the use of the partial rotation for 14-16 months, and an increase of the behaviors of play with the lever and of rotating and retrieving the toy for 16-24 months. This lever task has similarities with our tube task in its structure. Indeed, when the child does not directly solve the task, the exploration of one accessible part of the apparatus (the accessible side of the lever) can make the toy come closer, at which point the infant can directly grasp the toy. This behavior is categorized by the authors in one particular strategy called "operational preoccupation" (p795):

Step IV: operational preoccupation.-The operationally preoccupied child repeatedly rotates the bar away from his midline, but his preoccupation with rotation is such that he ignores the goal, not looking at the toy either during or after his rotations.

His repeated rotations often bring the goal to within reaching distance, but he does not take advantage of the closeness, seeming to ignore the goal object. (One child brought the goal object so close that it actually hit his opposite arm. Only then did he look at it, but he still did not take it.)

Occasionally, when the child is just about to repeat an operation, his attention may be directed to the area of the goal, as by a noise from the direction of the goal. When this happens, he is "freed" from preoccupation with operation of the lever, spots the goal, stops his repetitions, and, if the goal is close enough, tries to reach it directly.

The reported behaviors are similar to infants in the tube task inserting all objects into the tube, which as a side effect can bring the toy within reach, in which case some of them grasp and play with the toy, while other ignore it (Observation C2-1) or place the toy back into the tube continuing their action (A3-1, A3-2 and others). We can see here that the authors consider only one "goal", the one decided by the experimenter: getting the toy, or the toy itself, while other behaviors that ignore that goal are "preoccupations". However, those behaviors could well be the result of an intrinsic motivation to explore the environment, with the babies' goals and strategies sometimes aligned with the one of the experimenter, sometimes not. The children may "disregard the goal while getting enmeshed in the means" as the authors put it, because of attentional and processing constraints, but they may also continue to explore those means because they find them more interesting. In both this lever task and our tube task, the success rates could thus be driven by several factors: the tool-use skills, the interest in getting the toy, the diversity of exploration, and serendipity, among others.

We have shown that analyzing the behaviors of infants with a fine-grained time scale and taking into account all their actions, their gaze, and the related events in the environment can help us understand the interaction between intrinsic motivations and the task progress in such tool-use experiments. Our analysis has similarities with the microgenetic approach (Chen et al., 2000) developed to study changes in children thinking and behaviors. This approach emphasizes the importance of 1) observations spanning a time from before the period of rapid change to after the stable use of new thinking, 2) high density observations during this period relative to the rate of change, 3) trial-by-trial assessments of ongoing changes.

In Schauble (1996), the microgenetic method allowed to observe the structure of change in ways of thinking in a task where fifth graders and adults had to come up with experiments to understand the causal role of four variables. By repeating the task over multiple trials and sessions, they could show that the majority of belief updates were changed back and forth multiple times. In infants, Adolph et al. (1997) assessed on a trial-by-trial basis the changes in locomotion skills on surfaces of different slopes, from the point where babies started to crawl, to the point where they were proficient walkers. They show that babies do not transfer the skills they

learned to cope with slopes while their dominant locomotion strategy was crawling to the later situation where walking is their dominant strategy, but rather relearn strategies adapted to those slopes. More recently, DiMercurio et al. (2018) studied babies' spontaneous movements with the arm in the first 2 months of life, with dense recordings of the touches to the body and the environment. They found that infants self-generate many arm movements leading to hundreds of touches in a few minutes, while half of their time is spent moving from one touch location to another. They document the types of touches and the proportion of each type of touches, together with the evolution of those behaviors across sessions. In those experiments, dense observations allowed to give insights in the time course of behaviors, strategy use or learning.

In a tool-use experiment with 1.5 and 2.5 year olds where infants had to retrieve an out-of-reach toy with one of the six available tools (Chen et al., 2000), the time scale of observations was the trial. In their control condition, the mother just asked the child to get the toy. In the hint condition, the experimenter moreover suggested to use the target tool. Finally, in the modeling condition, the experimenter actively showed to the infant how to retrieve the toy with the target tool. The authors show that in the control condition only few children succeeded to retrieve the toy with the tool even after three problems. However, in the hint and modeling conditions, a large proportion of 1.5-year olds and most of the 2.5-year olds succeeded to use the tool strategy by the end of the experiment. With respect to the strategic variability, the authors measured that 74% of toddlers used at least three strategies. The different strategies observed were to lean forward and try to retrieve the toy with the hand, to grab one of the tool and try to catch the toy with the tool, to ask to the mother if she could retrieve the toy for them, or to walk around the table to look at the toy from different angles. They document the changes in use of each strategy depending on trial and age, and argue in favor of the overlapping waves theory stating that each child uses a diverse set of strategies at any point in time and that their proportion of use do not change abruptly (Siegler, 1996). In our study, we showed that the observation of infant's behavior with a higher density than the trial can give insights on infant motivations and behaviors in particular in tool-use experiments.

The extensive annotations of behaviors and events in the tool-use experiment allowed to picture interesting correlations and interactions between the baby and its environment. However, the choice of particular correlations to test was made through a post-hoc exploratory analysis of the data, so that those results would benefit confirmation in a similar tool-use experiment. Also, those statistical tests, and in particular the comparisons between different training conditions, may have suffered from the relatively small number of analyzed samples (32 babies).

We also annotated the apparent goals and associated strategies of babies, from a more subjective point of view than actions and gaze. If a particular behavior is correlated with a particular goal, it may be because the babies display more that behavior as they pursue this goal, but it could also be that the observer thinks

the baby is following that goal because that goal is thought to correspond to this behavior. Moreover, it was often hard to distinguish between several possible goals and strategies, such as between the goal of retrieving the toy with the tool-use strategy and the goal of inserting all objects. In those cases we annotated all the goals compatible with the behaviors, which may be a reason why it was hard to observe differences when comparing behaviors depending on goal.

The tube task only had one particular salient goal, and one type of strategy to reach that goal. In order to study intrinsically motivated exploration, one could design an apparatus with more degrees of freedom related to tool use: with several salient goals, with different solutions to each tool-use problem, and potentially with different possible difficulties in a same setup. In Hoicka et al. (2013), the authors define a test for evaluating creativity in young children (called “divergent thinking”), with an interesting apparatus used to trigger spontaneous exploration. Children are free to explore a box with many different parts, together with independent toys. The test measures the number of different interactions between the toys and the setup parts, and is shown to correlate with other usual tests of creativity, and to work with infants as young as 19 months. For testing spontaneous exploration related to tool use, we can imagine a box with several toys somehow locked inside the box, and many different ways to unlock those toys. For instance, the Multi Access Box Paradigm of Auersperg et al. (2011) allows multiple tool-use solutions to the problem of retrieving food in bird studies. Also, to facilitate coding, such a tool box could integrate digital sensors recording the movements of the toys and parts of the box.

To conclude, we have shown that the goals and strategies of babies were diverse in this tool-use task, and were mostly unexpected by the time of the design of the apparatus. The spontaneous exploration of many goals and strategies can lead to the discovery and learning of particular strategies to solve particular goals, so that intrinsic motivations seem to play a key role in the exploration of such tool-use setups and therefore can interfere with the measures of success and learning. However, most questions have just been raised and left mostly unanswered: how do babies select their goal and their strategy at any point in time? How does this choice depend on their previous experience with the setup and the progress to achieve goals? How do extrinsic factors such as an experimenter or caregiver attracting attention interplay with intrinsic motivations?

Computational and robotic modeling, by providing a platform to run and evaluate the behaviors of artificial agents endowed with particular forms of motivations could help study some of those questions. If we want to model the behavior of babies in tool-use experiments, we cannot ignore that they pursue their own goals and strategies, while being constrained by limited attentional and processing abilities and by external guidance. By testing different algorithmic implementations of intrinsic motivations, we could observe the behaviors induced by each algorithm, study in detail the interaction of the curiosity component with the environment variables, and refine our hypotheses about the mechanisms of curiosity in children. In Chapter 4,

in an attempt to answer some of the previous questions we design several robotic models of curiosity-driven exploration and evaluate them in simple tool-use setups. In Chapter 5, we tackle the problem of speech development together with tool-use development in a same unified framework.

Chapter 4

Modeling Intrinsic Motivations in the Development of Tool Use

Summary

Tool use is a fundamental ability displayed by animals and humans who start to manage sticks, rakes and spoons in their first two years. The understanding of the learning and development of tool use is a central question in cognition research. Babies are incredible curious explorers of their environment, from exploring their body to playing with objects and combining them. Previous work showed that intrinsic motivations could allow the emergence of naturalistic developmental trajectories in robotic models of simple aspects of sensorimotor development. The mechanisms of intrinsic motivations in the exploration of tool-use precursors and their potential role in the development of tool-use skills are open for debate. In this chapter, we study how the particular implementations of intrinsic motivations to self-generate interesting goals together with the particular representation of goals can play a role in the tool-use progression in a robotic model. In a first experiment (Forestier and Oudeyer, 2016a), we study the evolution of behaviors of robotic agents depending on the intrinsic motivation and the representation of the environment. We show that an intrinsic motivation based on the learning progress to reach goals with a modular representation can self-organize phases of behaviors in the development of tool-use precursors that share properties with child development. In a second experiment (Forestier and Oudeyer, 2016c), we focus in particular on the choice of tool-use strategy with a model of the child experiment of Chen et al. (2000). We show that particular implementations of intrinsic motivations are compatible with their observations, but are usually not considered in the interpretation of those experiments.

The understanding of tool-use development in young children is one of the key questions for the more general understanding of the ontogeny of human cognition. Indeed, a series of abilities are progressively developed from the simplest reaching movements of the arms through more dexterous manipulation of a spoon, towards advanced control of multiple interacting objects. The latter shows an understanding of shapes, forces and other physical properties that can be hierarchically recruited for mental transformations and planning operations which are pillars of human cognition. Child development has first been described as staircase-like successive stages in which children go through (Piaget, 1952). More recently, other views were developed to describe the structure and variability of children's observed developmental paths. In particular, the development of tool-use precursors can be described as three consecutive and overlapping stages of behaviors where sequential learning and goal-directed behaviors play an increasing role (Guerin et al., 2013): body babbling, behaviors with a single object, and behaviors with several interacting objects. For example, a study of free play (Zelazo and Kearsley, 1980) shows that at $9\frac{1}{2}$ months play is mostly composed of tactile examination, waving or mouthing of a single object but simple relational acts of banging two objects together are already present. Later at $13\frac{1}{2}$ months, the study reveals that most children instead prefer to explore the relationships among objects, but still show behaviors of the previous phase. Siegler describes the behavioral structure in a child's set of current methods to solve a problem in the overlapping waves framework (Siegler, 1996), which allows to represent the variability in the choice of strategies and the evolution of strategy preferences.

Intrinsic motivations, sometimes called "curiosity", have recently been suggested to play an important role in driving exploration and learning in infants (Gottlieb et al., 2013; Kidd and Hayden, 2015). Intrinsic motivations have been defined as mechanisms that push infants to explore activities for their own sake, driven by the search of novelty, surprise, dissonances or optimal challenge (see Background chapter). However, intrinsic motivations are most often not considered in the interpretation of results from psychological experiments. In the previous chapter, we analyzed a tool-use experiment with 21-month olds originally aiming at assessing a particular sequential tool-use skill, where a toy was placed in the middle of a transparent tube and the infant had to retrieve the toy by inserting several blocks into the tube. We found that infants could spontaneously generate their own goals and pursue them despite the fact that the particular goal of retrieving the toy was made salient in the experimental setup and that the attention of the infants was driven towards that goal. The intrinsic motivation to explore and learn during this task may have interfered with the success scores for solving the task as the exploration of alternative goals (such as inserting all blocks into the tube) could lead to the accidental discovery of a strategy to solve the task. We argued that intrinsic motivations could play a key role in infants' exploration and learning even during "test" tasks where the goal of the subject in the task seems obvious to the experimenter, and should be considered in the interpretation of such experiments.

In order to understand the mechanisms of tool-use development in infants and in particular how they discover, explore and learn through a combination of intrinsic motivations and extrinsic factors, we now follow the approach of computational and robotic modeling. By implementing intrinsic motivations in artificial agents and evaluating their behaviors in tool-use environments, robotic models allow to test alternative hypotheses about the precise mechanisms of exploration and learning in tool use.

In the last decade, various families of computational models of intrinsic motivation were developed, often based on the formal frameworks of active learning and reinforcement learning (Baldassarre and Mirolli, 2013). One family of models, that has targeted to study the developmental dimensions of intrinsic motivation, has considered a curiosity-driven learning mechanism where the learner actively engages in sensorimotor activities that provide high learning progress, avoiding situations that are too easy or too difficult and progressively focusing on activities of increasing complexity (Gottlieb et al., 2013). Such computational models have shown that developmental trajectories could emerge from the curiosity-driven learning of sensorimotor mappings, in very different settings. In the Playground Experiment (Oudeyer et al., 2007), a quadruped robot motivated to maximize its learning progress acquired how to use its motor primitives to interact with the items of an infant play mat and a robot peer, following a self-organized learning curriculum. In Baranes and Oudeyer (2013), such mechanisms were shown to allow for an efficient learning of large repertoires of skills involving high-dimensional continuous actions, as intrinsic motivation guided the system to explore sensorimotor problems of increasing complexity. In a model of active vocal development (Moulin-Frier et al., 2013), an agent had to learn how to produce sounds with its vocal tract by self-exploration combined with imitation of adult speech sounds. This model reproduces major phases of infant vocal development until 6 months. In both studies, developmental trajectories are emerging from learning, with both regularities in developmental steps and diversity.

Existing models have considered the exploration and learning of sensorimotor correspondences mapping a motor space to a single task/sensory space. However, in the perspective of an open-ended development of reusable skills, and specifically in the development of tool use, multiple interdependent and organized task spaces should be available to the agent. For instance, using a tool to act upon an object could make use of previously explored interaction with the tool. An intrinsic motivation towards learning progress maximization could particularly be useful in the context of tool use where progress on some high-level task can not happen before progress on lower-level tasks have been made, by focusing training on currently learnable self-generated tasks.

In this chapter, we extend those curiosity-driven exploration models to be able to leverage the sensorimotor structure of tool-use robotic environments. We hypothesize that several mechanisms play a role in the progression between phases of tool-use behaviors, in particular 1) the intrinsic motivation to explore through a self-regulation

of the growth of complexity of self-selected skills or tasks; 2) the structure of the representation used to encode sensorimotor experience.

In a first experiment, we study the evolution of behaviors across the learning of tool-use precursors, depending on properties of the intrinsic motivation component and of the learning representation (Section 4.1). In a second experiment, we model one of Siegler’s tool-use experiment with babies to focus on the evolution of the use of set of strategies to solve a tool-use problem (Section 4.2).

4.1 Developmental Trajectories in Tool Use

In this section, we study the role of intrinsic motivations and of environment representations in the curiosity-driven exploration of a tool-use environment. We try to understand how particular implementations of intrinsic motivations and environment representations can modulate the exploratory behaviors and the learning of tool-use skills and how the behaviors of the artificial agent can fit the ones of infants in tool-use development. To this end, we design a tool-use robotic environment where a 2D articulated arm with three joints plus a gripper can grab one of two available tools to move an out-of-reach toy. In such environments, several skills need to be learned related to tools and toys, such as controlling the hand and gripper of the robot, reaching for the tools, and controlling the toy with the tools. We define a set of modular sensory spaces that structures the observations from the environment to reflect the interaction of the different items of the tool-use environment.

We introduce the HACOB exploration architecture (Hierarchical Active Curiosity-driven mOdel Babbling) that leverages this sensorimotor modularity to efficiently learn the tool-use skills with an autonomous developmental trajectory. We compare several implementations of intrinsic motivations with a different choice of sensorimotor model to explore: random, or based on the learning progress. We also assess different representations of the environment, hierarchical or flat.

In the different learning conditions, we simulate many independent agents to study the typical evolution of behaviors during exploration. We show that overlapping phases of behaviors are autonomously emerging for agents using an intrinsic motivation based on learning progress and a modular representation with a hierarchy of models.

4.1.1 Methods

Environment

We simulate¹ a 2D robotic arm that can grasp tools that can be used to move an object into different boxes in the environment. In each trial, the agent executes a

¹Source code and notebooks available as a GitHub repository at <https://github.com/sebastien-forestier/CogSci2016>

motor trajectory and gets the associated sensory feedback. Finally the arm, tools and objects are reset to their initial state. The next sections precisely describe the items of the environment and their interactions. See Fig.4.1 for an example state of the environment.

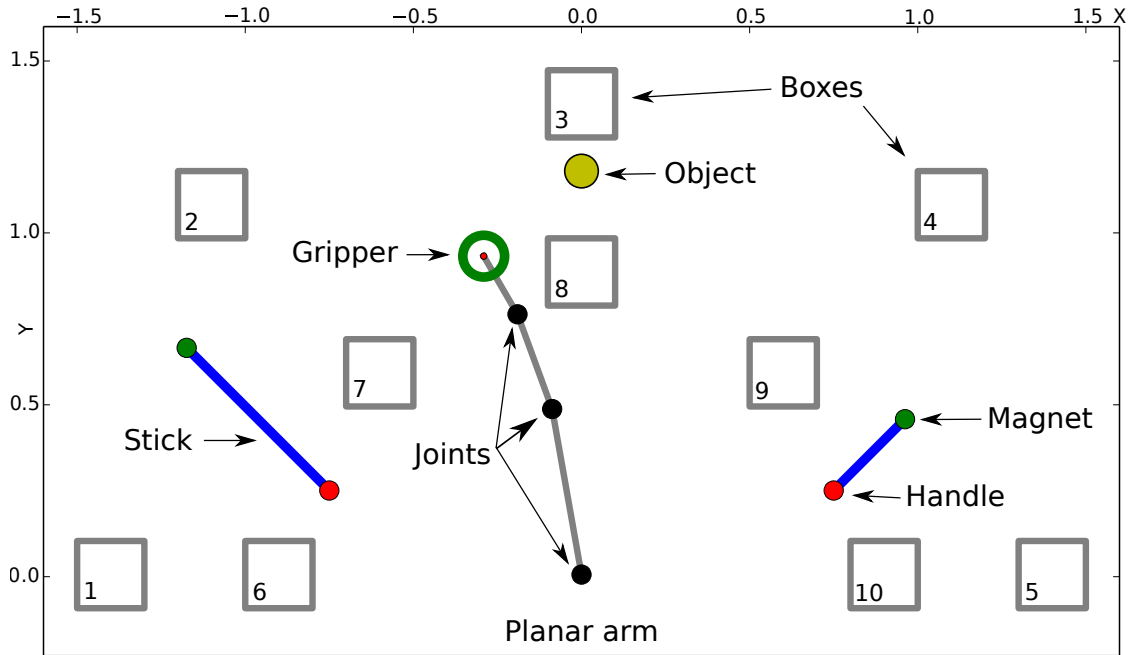


Figure 4.1: Example state of the environment.

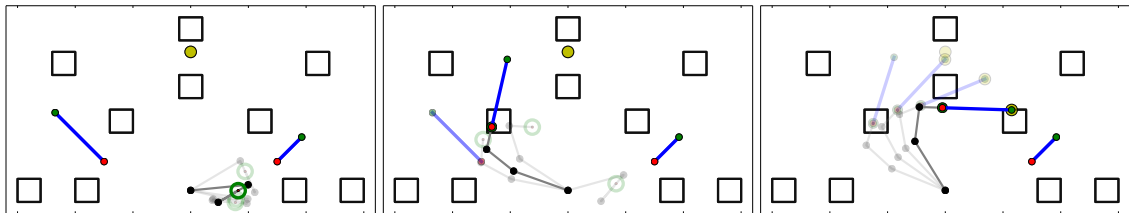


Figure 4.2: Example arm trajectory: position of the arm at time steps 17, 33 and 50, with several intermediate positions, along the 50 steps movement.

Robotic Arm The 2D robotic arm has 3 joints plus a gripper located at the end-effector. Each joint can rotate from $-\pi$ rad to π rad around its resting position, mapped to a standard interval of $[-1, 1]$. The length of the 3 segments of the arm are 0.5, 0.3 and 0.2 so the length of the arm is 1 unit. The resting position of the arm is vertical with joints at 0 rad and its base is fixed at position $[0, 0]$. The gripper g

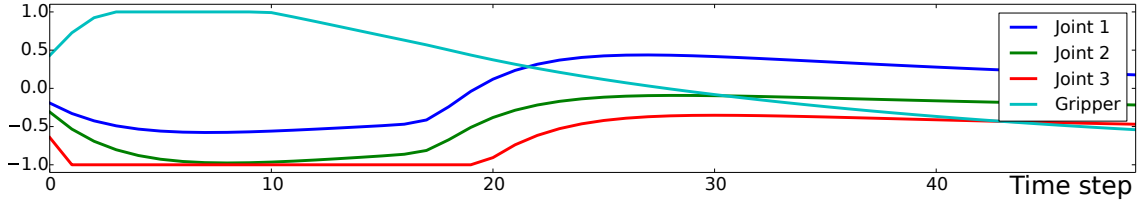


Figure 4.3: Trajectory of each of the four virtual motors, generated by a DMP.

has 2 possible positions: *open* ($g \geq 0$) and *closed* ($g < 0$) and its resting position is *open* (with $g = 0$). The robotic arm has 4 degrees of freedom represented by a vector in $[-1, 1]^4$. A trajectory of the arm will be represented as a sequence of such vectors.

Motor Control We use Dynamical Movement Primitives (Ijspeert et al., 2013) to control the arm’s movement as this framework permits the production of a diversity of arm’s trajectories with few parameters. Each of the 4 arm’s degrees-of-freedom (DOF) is controlled by a DMP starting at the rest position of the joint. Each DMP is parameterized by one weight on each of 2 basis functions and one weight specifying the end position of the movement. The weights are bounded in the interval $[-1, 1]$ and allow each joint to fairly cover the interval $[-1, 1]$ during the movement. Each DMP outputs a series of 50 positions that represents a sampling of the trajectory of one joint during the movement. The arm’s movement is thus parameterized with 12 weights, represented by the motor space $M = [-1, 1]^{12}$.

Objects and Tools Two sticks can be grasped by the handle side in order to catch an out-of-reach object. A small stick of length 0.3 is located at position $(0.75, 0.25)$ and a long stick of length 0.6 is located at position $(-0.75, 0.25)$ as in Fig. 4.1. An object (yellow ball), initially at position $(0, 1.2)$, can be caught by the magnetic side of one of the two sticks, moved and possibly placed into one of ten fixed squared boxes. If the gripper is closed near the handle of a stick (closer than 0.2), it is considered grasped and follows the gripper’s position and the angle of the arm’s last segment until the gripper opens. Similarly, if the magnetic side of a stick reaches the ball (within 0.1), the ball will then follow the magnet. The ten boxes (identified from 1 to 10) are static and have size 0.2. Boxes 1 to 5 can only be reached with the long stick, and the other five boxes can be reached with both sticks.

Sensory Feedback At the end of the movement, the robot gets sensory feedback from the different items of the environment (S , 25D). First, the trajectory of the gripper is represented as the x and y positions and the aperture (1 or -1) of the gripper at 3 time points: steps 17, 33, 50 during the movement of 50 steps (S_{Hand} , 9D). Similarly, the trajectories of the end points of the sticks are 3-point sequences of x and y positions (S_{Stick_1} and S_{Stick_2} , 6D each). It also gets the position of the single

object at the end of the movement ($S_{Object}, 2D$). The agent receives the identifier (from 1 to 10) of the reached box if one of them has been reached by the ball, 0 otherwise. It also receives the distance between the ball at the end of the movement and the closest box ($S_{Boxes}, 2D$).

Learning Architectures

The problem settings for the learning agent is to explore its sensorimotor space and collect data so as to discover how to produce a diversity of effects, and to learn repertoires of skills allowing to reproduce these effects in the form of inverse models. Consequently, the system is not given a priori a single target task to be solved: it rather autonomously selects the sensorimotor problems it will focus on through an intrinsically motivated selection of sensorimotor models.

Flat Architectures We define a flat architecture as directly mapping the motor space M (12D) and the sensory space S (25D). To do so, the agent needs a sensorimotor model that learns the mapping and provides inverse inference of a probable m to reach a given s . The sensorimotor model stores new information of the form (m, s) with $m \in M$ being the experimented motor parameters and $s \in S$ the associated sensory feedback. It computes the inverse inference with the nearest neighbor algorithm: it gets the motor part of the nearest neighbor in S of the given s , and adds exploration noise (Gaussian with $\sigma = 0.01$) to explore new motor parameters.

The agent also needs an interest model that chooses goals in the sensory space. The control condition is a random motor babbling condition (F-RmB) that always randomly chooses new motor parameters m . In the other conditions, the agent performs Goal Babbling, a method by which it self-generates goals in the sensory space and tries to reach them. To generate those goals, different strategies have been studied (Baranes and Oudeyer, 2013). It was shown that estimating the learning progress in different regions of the sensory space and generating the goals where the progress is high leads to a fast learning. However, this cannot be applied in a 25D sensory space as a learning progress signal cannot be estimated in this volume. Thus, in the flat random goal babbling condition (F-RGB), we use a random generation of goals in the sensory space, which was nevertheless proven to be highly efficient in complex sensorimotor spaces (Rolf et al., 2010).

Hierarchical Architectures The 25D sensory space can be structured to reflect the interaction of the different items of the environment. Indeed, the arm motor position influences the gripper, which influences one of the tools (but not both at the same time), which influences the position of the object and the filling of the boxes. We thus study here learning architectures that could make use of this sensorimotor structure, and we call them hierarchical. Those architecture learn 6 models at the same time (see Figures 4.4 and 4.5: gray squares are models). Each of those models

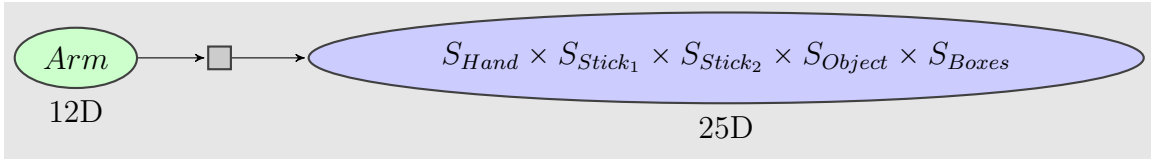


Figure 4.4: Flat

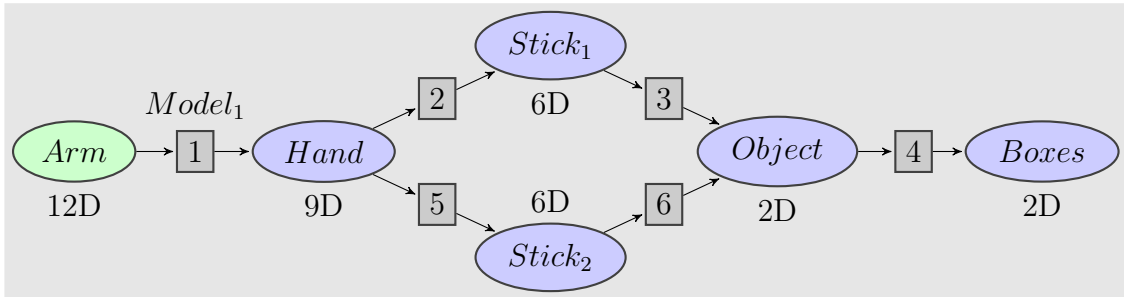


Figure 4.5: Hierarchical

functions in the same way as the random goal babbling flat architecture (F-RGB). Each model has a motor space (e.g. motor space of model 2 is S_{Hand}), a sensory space (respectively S_{Stick_1} , see arrows in Fig. 4.5), and can choose goals randomly in this sensory space. At each iteration, the architecture first has to choose the model in which to pick a goal, a procedure that we call Model Babbling. Once a model is chosen, it finds a goal in its sensory space, and infer motor parameters (that can be in the sensory space of a lower-level model) to reach that goal. Then, it passes those parameters as a goal to a lower-level model, which similarly infers motor parameters and passes those ones until the actual *Arm* motor space gets parameters to execute in the environment (with the same exploration noise as in Flat architectures). Model 4 has also to choose which lower-level model to use in order to reach an object end position s_o in S_{Object} , as two models (3 and 6) have S_{Object} as sensory space. Model 4 chooses the tool that enabled reaching s_o as close as possible in the past, e.g. if model 3 has in its history a reached sensory point s closer to s_o than any reached point with model 6, then model 3 is chosen. Finally, when motor parameters m are executed in the environment and feedback s is received, the mappings of all models are updated. However, only the tool-particular models are updated when a tool is currently held.

Random vs Active Model Babbling In a first condition, the agent randomly chooses the model that will find a goal, which is called Random Model Babbling (H-RMB). The problem of Model Babbling is an instance of strategic learning (Nguyen and Oudeyer, 2012), where different outcomes and strategies to learn them are available and the agent learns which strategies are useful for which outcomes. In that

paper, they show that an active choice of the outcomes and strategies based on the learning progress on each of them increase learning efficiency compared to random choice. To develop active learning strategies, we first define a measure of learning progress for each of the 6 models. When a model has been chosen to babble, it draws a random goal s_g , and finds motor parameters m to reach it using the lower-level models. The actual outcome s in the sensory space of the model, associated to m might be very different from s_g as this goal might be unreachable, or because lower-level models are not mature enough for that goal. We define the competence associated to a goal s_g as the negative distance between the goal and the reached point, divided by the maximal distance in this space, to scale this measure across different spaces:

$$C(s_g) = -\frac{\|s_g - s\|}{\max_{s_1, s_2} \|s_1 - s_2\|} \quad (4.1)$$

and the interest $I(s_g)$ associated to this goal as

$$I(s_g) = |C(s_g) - \text{mean}_{kNN}C(s_g)| \quad (4.2)$$

where $\text{mean}_{kNN}C(s)$ is the mean competence of the ($k = 20$) nearest previous goals (k-Nearest Neighbors algorithm). The interest of a model is initialized at 0 and updated to follow the interest of the goals (with rate $n = 200$):

$$I_{model} = \frac{n-1}{n} I_{model} + \frac{1}{n} I(s_g) \quad (4.3)$$

In condition H-P-AMB, the choice of model is probabilistic and has $\epsilon = 10\%$ chance to be random, and $(1 - \epsilon)$ to be proportional to their interest. In condition H-GR-AMB, the choice of model is greedy (model with maximum interest) but also with $\epsilon = 10\%$ of random choice. Finally, condition H-P-AMB-CTC (Curiosity-driven Tool Choice) is the same as H-P-AMB but the choice of the tool to use (model 3 or 6) is made with probabilities proportional to the interest of the two models, instead of being based on the more competent tool for the given object goal position. We call HACOBA this Hierarchical Active Curiosity-driven mOdel Babbling algorithmic architecture with the algorithms H-P-AMB and H-P-AMB-CTC being two variants of the architecture.

4.1.2 Results

We perform 100 independent simulations of 100000 iterations per condition, starting with 100 iterations of motor babbling. Fig. 4.6 shows details about one trial of the condition H-P-AMB. We can see the interest of each model during one simulation, and the corresponding explored object space. The interests of models 2 and 5 increase once the arm succeeded to grab the corresponding stick. Following that, the interests of models 3 and 6 increase once the object has been reached.

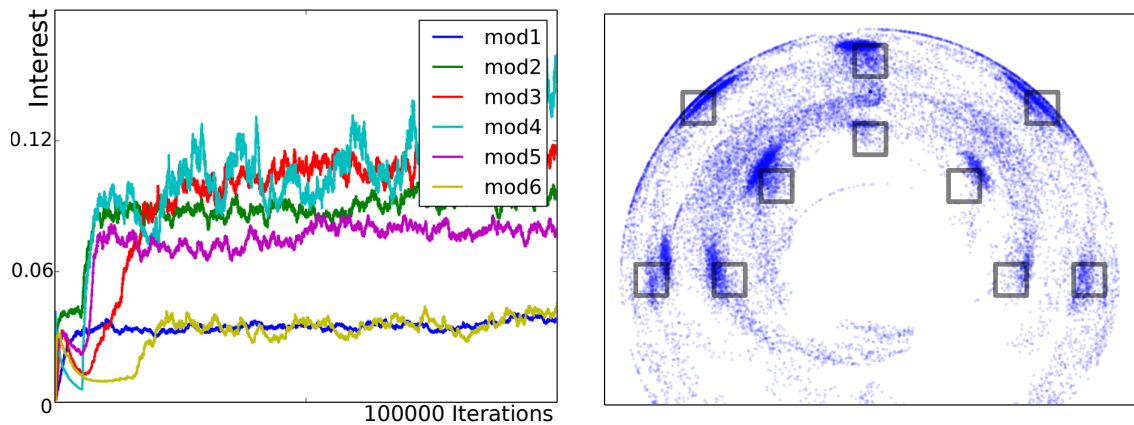


Figure 4.6: Condition H-P-AMB. Left: Interest of each model. Right: Exploration of the object space: each dot is the position reached with the object at the end of a movement.

Behavioral Evolution and Exploration Efficiency

We provide a measure of three types of behaviors with objects during exploration. In the first category (*hand*) the arm did not grab any stick and thus did not move the out-of-reach object. In the second category (*stick*), the arm did grab one of the two sticks but did not touch the object with it. The third category (*object*) contains the movements where both a stick was grabbed and the object was moved by the stick. Fig. 4.7 shows a typical evolution of the proportion of the three categories of behaviors. We performed a more detailed analysis (see Table 4.1) by counting the trials where the evolution of the behaviors were similar to the ones found in infant development of the interaction with object (Guerin et al., 2013). A structure was considered similar to infant behavioral structures if it validated each of the following criteria: behaviors of categories *stick* and *object* increase from 0 to more than 10% (potentially after an initial phase with a steady low value), are followed by a curve with small slope and no abrupt changes, and behaviors of category *object* start to raise at least 1000 iterations after *stick* started to raise (see Fig. 4.7(c) for a valid instance). Also, the median number of abrupt changes across trials are reported in Table 4.1 (as the sum of steady changes of more than 10% in the three behaviors), with a significant difference between condition H-GR-AMB and others (Mann-Whitney U tests, $p < 10^{-4}$).

For each condition we also measured the total exploration of the sensory spaces during training. The exploration of the hand, sticks and object spaces is defined as the number of reached cells in a 100×100 discretization of the (X,Y) space of their position at the end of the movement. Boxes' exploration is the number of boxes reached with the object during training. Fig. 4.8 shows the exploration of the different sensory spaces for each condition. We provide Mann-Whitney U test results

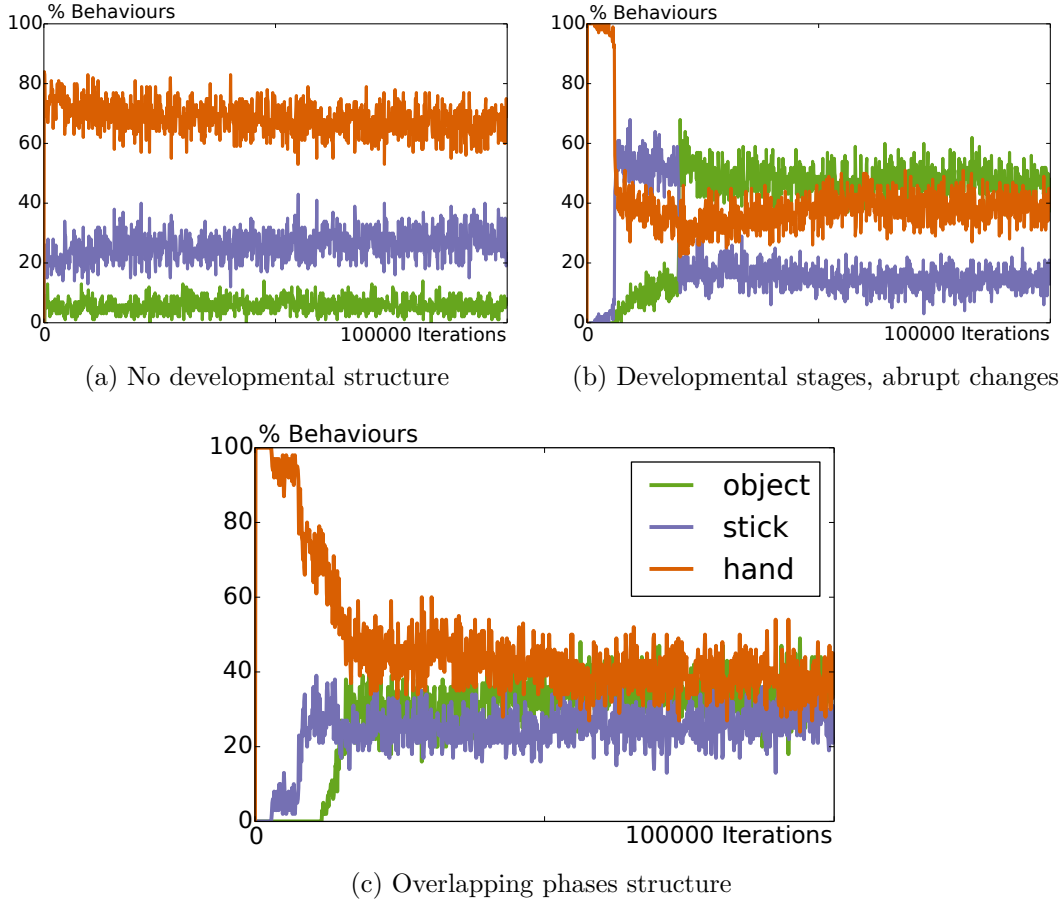


Figure 4.7: Typical behavioral evolution in the conditions (a) F-RGB, (b) H-GR-AMB, (c) H-P-AMB.

of comparisons of total exploration for some pairs of conditions. One star means $p < 0.05$, two: $p < 10^{-2}$, three: $p < 10^{-3}$, four: $p < 10^{-4}$.

Evolution of Strategy Preferences

Finally, we compare the structure of tool choice made to reach object goal positions in two conditions for which only this choice differs. Fig. 4.9 shows the choice of tool to reach a given object goal position in the conditions H-P-AMB and H-P-AMB-CTC. When model 4 is babbling, it infers the best object position s_o to reach a random goal $s_b \in S_{Boxes}$. We plot all the choices that model 4 made during exploration, at position s_o on a 2D space, with color blue if $Stick_1$ was chosen and red if $Stick_2$ was chosen. In condition H-P-AMB, we can see strong boundaries between tool choice regions. By contrast, in condition H-P-AMB-CTC, both tools are chosen in all regions.

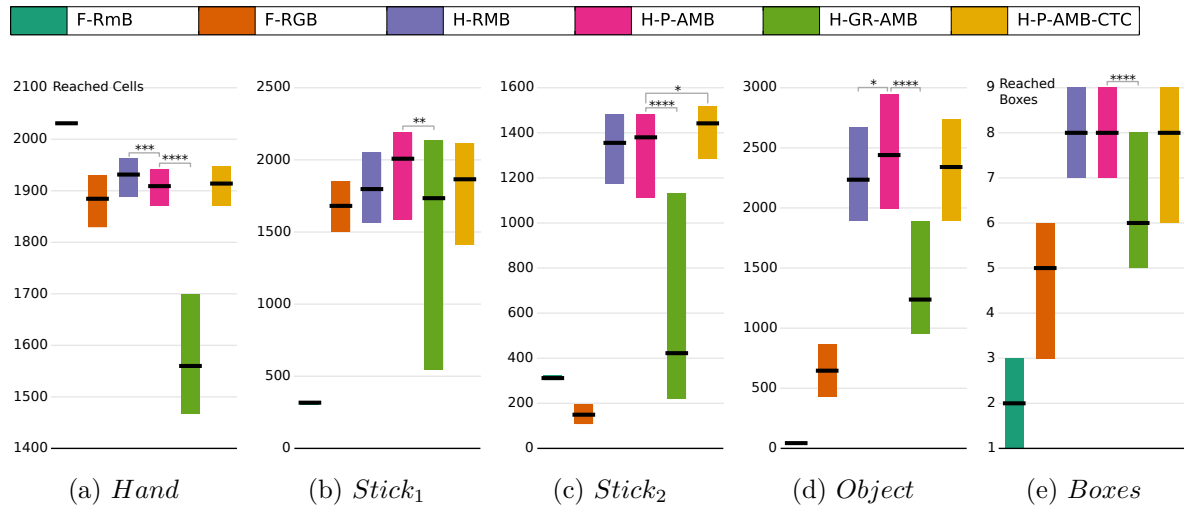


Figure 4.8: Exploration of sensory spaces. Box plots show medians and quartiles of the 100 trials.

Table 4.1: Behavioral results

Condition	Number of Trials validating criteria	Median number of Abrupt changes
F-RmB	0	0
F-RGB	0	1
H-RMB	60	2
H-P-AMB	70	2
H-GR-AMB	7	6
H-P-AMB-CTC	79	1

4.1.3 Discussion

Behavioral Evolution

The results show different structures of behavior evolution in the different conditions. Flat architectures cannot efficiently learn in this environment with a high-dimensional sensory space. Therefore, they do not show structure in the behavioral evolution but rather steady proportions of the three behaviors. By contrast, hierarchical condition H-GR-AMB shows successive behavioral steps with abrupt changes, which reflects the greedy choice of model to babble. When one model becomes more interesting than another, it is chosen for a large number of iterations until another model exceeds its interest. Random model babbling shows overlapping phases structures more compatible with infants' studies in the evolution of the three behaviors, but less than active model babbling (60% instead of 70% or 79%). This is because random model

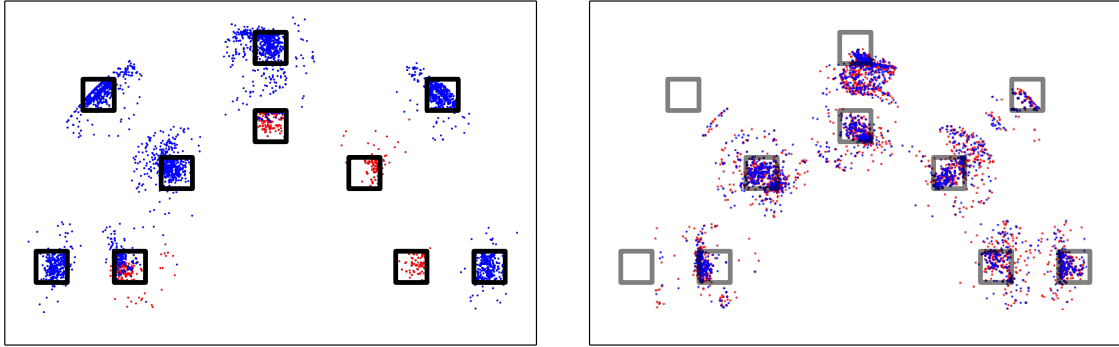


Figure 4.9: Chosen tool depending on object goal position. Blue: long stick choice. Red: small stick. Left: H-P-AMB, strong boundaries between tool choice regions. Right: H-P-AMB-CTC, parallel exploration of both tools in all regions.

babbling does not adapt its choice of models to their interests along development. Indeed, it often explores model 1 even if it is sufficiently explored to make progress on higher-level models, and so explores less the object position space than active model babbling (H-P-AMB). Also, all models are still useful to explore after the number of iterations simulated here so the first behavioral phases (*hand* and *stick*) do not lessen towards the end of the simulations in condition H-P-AMB.

Different Tools to Reach a Toy

The comparison of conditions H-P-AMB and H-P-AMB-CTC shows that when the agent chooses the tool to reach a given object goal position based on the interest of the corresponding models, both tools are trained to reach all goals instead of training only the best performing tool. Indeed, with the active curiosity-driven choice of tool, the small stick has produced more diverse effects on the object than in the optimal tool's condition (Fig. 4.8c), even if those effects could also have been generated with the long tool. The behaviors emerging from this curiosity-driven tool choice are in accordance with Siegler's overlapping waves theory. Indeed, Siegler describes the use of strategies in infants and explains that non-optimal strategies continue to be explored as they might turn out to be good ones in the end, for this problem or for related problems.

To our knowledge, HACOB is a first model of the curiosity-driven development of tool use, and first to show the autonomous emergence of overlapping phases in the development of simple tool use in a simulated robotic setup. This model also accounts for the intrinsically-motivated parallel exploration of different tools to reach one goal, in line with Siegler's overlapping waves theory. Other models predefine successive phases in object affordances learning (Ugur et al., 2015), do not study the role of intrinsic motivation in tool affordances learning (Stoytchev, 2005), or have only considered the autonomous development of single object manipulation (Gottlieb

et al., 2013).

In artificial agents, the intrinsic motivation properties and implementation seem to influence the exploration of non-optimal strategies. In the next section, we focus in more details on this question through experiments in an environment where several different strategies (reaching with the hand or with a tool) can be used to solve the same set of problems (reaching a toy).

4.2 Overlapping Waves of Strategy Preferences

Siegler's overlapping waves theory (Siegler, 1996) describes and models the way infants represent and select a set of currently available methods to solve a problem. According to this theory, infants maintain their set of methods (also called strategies) with associated frequencies depending on the past history of the use of those methods. The frequencies evolve over time while new strategies are discovered which explain the observed changes in behavior. For instance, when learning the mathematical addition, infants use different methods from one trial to another, and may continue to use non-optimal methods for a long period of time even if they already know more efficient methods. Siegler suggested that such continued exploration of alternative and sub-optimal methods to solve a family of problem may be useful to acquire skills that will later facilitate the resolution of new problems. This cognitive variability could be an essential mechanism to acquire greater knowledge, which might be more important for learning in childhood than just having high quality performances on specific tasks.

Siegler and colleagues developed several computational models of strategy selection and evolution to account for how children learn how to add integer numbers: ASCM (Adaptive Strategy Choice Model, Siegler (1996)), and SCADS (Strategy Choices and Strategy Discoveries, Shrager and Siegler (1998)). Those models are argued to closely parallel the development of addition strategies with the use of several strategies, with errors in the execution of those strategies. In SCADS, furthermore, a mechanism allows the discovery of new strategies and the authors show that the same strategies are discovered and in the same sequences as with children. In the two models, the strategies are selected with frequencies that are directly proportional to (called a "matching law" on) their success rate in the corresponding previous problems. This model also included a novelty bias to explore new strategies more than their success rate would allow: the value for exploring new strategies was initialized optimistically (then decreasing in time if success rate did not rise). The focus of this model has been the mode of strategy selection (matching law), with a measure of the value of strategies based on their performance to solve a given task. However, these models have not considered other forms of motivations, such as curiosity-driven exploration, which as we suggested (see previous chapter) could play an important role in learning.

In the context of tool-use development, Chen et al. (2000) conducted an experiment

with 1.5- and 2.5-year-olds that had to retrieve an out-of-reach toy with one of the six available tools. Children were exposed to a sequence of three similar problems with different tool shapes and visual features, but for each problem only one tool was effective to retrieve the toy. They designed three conditions. In the control condition, the mother just asked the child to get the toy. In the hint condition, the experimenter moreover suggested to use the target tool. Finally, in the modeling condition, the experimenter actively showed to the infant how to retrieve the toy with the target tool. First, they show that in the control condition only few children succeeded to retrieve the toy with the tool even after three problems (less than 10% of the 1.5-year-olds and less than 20% of the 2.5-year-olds). However, in the hint condition and modeling conditions, a large proportion of 1.5-year-olds and most of the 2.5-year-olds succeeded to use the tool strategy by the end of the experiment. With respect to the strategic variability, the authors measured that 74% of toddlers used at least three strategies. The different strategies observed were to lean forward and try to retrieve the toy with the hand (forward strategy), to grab one of the tool and try to catch the toy with the tool (tool strategy), to ask to the mother if she could retrieve the toy for them (but she was told not to) or to walk around the table to look at the toy from different angles (indirect strategy), and finally some of the children did not engage in any of those strategies (no strategy).

Chen et al. (2000) reported the dynamics of strategy choice as an average over children. They showed that the tool strategy frequency was on average increasing with the successive trials and the forward strategy was decreasing in the hint and modeling conditions, whereas in the control condition the tool strategy remained stable. This pattern was interpreted by the authors as a clear overlapping waves pattern besides the fact that it was a pattern of the average over children. The overlapping waves theory suggests that this pattern of strategy change should be visible on a per child basis, meaning that each child should always use a set of strategies and smoothly change their frequency use. However, the observed average pattern does not imply that each child (or most of them) display an overlapping waves pattern. It could be that in Chen and Siegler's experiment, each child begins with the forward strategy, and at some point in the experiment (different for each child), switch to the tool strategy and never uses again the forward one. In that case, an average of the strategy use would also show a smooth increase in the tool strategy and decrease in the forward strategy use. Nevertheless, the authors also reported a measure that could disentangle the different hypothesis (Chen et al., 2000, p42). They measured the average proportion of trials where children used other strategies than the tool strategy after the first trial where they used the tool strategy. The toddlers in the control condition did use the other approaches than the tool strategy on more than half the trials after the first time they used the tool strategy (84% of the trials for 1.5-year-olds, 48% for 2.5-year-olds). In contrast, in the hint and modeling conditions, the young infants used other approaches in around 20% of the trials, and older infants in only 4%. This result showed that strategic variability did

continue after children began to use the tool strategy in the control condition but not in the hint and modeling conditions. Therefore, we do not agree with the conclusions of the authors saying that a clear overlapping waves pattern was visible regarding the change in forward versus tool strategy use. According to this analysis, overlapping behaviors were observed in this experiment only in the control condition where the mother just asked the infant to retrieve the toy, and the experimenter did not add further incentive.

In this section, we consider the problem of the modeling of overlapping waves of behaviors in the context of tool use. We will target to model alternative mechanisms that could be at play in Chen and Siegler’s experiment. The same model will be used for both free play exploration/learning of tool use (modeling learning of tool use taking place “at home” during the months preceding the lab sessions) and for exposure to evaluation in lab sessions with an incentive to solve a task. Indeed, a source of difficulty to interpret the results of behavioral experiments in babies is that it is difficult to control for what happened before the lab sessions. In particular, we can’t know exactly how much prior experience the toddlers had playing with objects and tools at home, what kind of tools were available, and how the caregivers were interacting with the child or answering its requests to get toys. Furthermore, understanding how the object saliency and the cues of the caretaker are interpreted by the children is an open question. The interpretation of these experiments has implicitly assumed that the experimental setup was designed so that the children would “want” to catch the toy (this also applies to similar experiments such as Fagard et al. (2016)). However, as we will suggest through the model below, alternative hypotheses can be considered (and be non-exclusive). In particular, we will suggest that a salient object may trigger curiosity-driven exploration, where the child explores to gain information about which strategy allows to get it (rather than trying to maximize its probability to actually catch it).

We build upon our previous model of the curiosity-driven development of tool use in a simulated 2D environment with objects and tools (see previous section, and Forestier and Oudeyer (2016a)). The agents in this experiment are learning several sensorimotor models structured in a hierarchy that represents the environmental structure. The use of an intrinsic motivation for the exploration of sensorimotor mappings yielding a high learning progress allowed the emergence of a smooth progression between overlapping phases of behavior similar to the one found in infants (Guerin et al., 2013). The intrinsic motivation self-organized a first phase where the agents were mainly exploring movements of the arm without touching objects, then the exploration of the interaction with a single object, and finally a smooth shift towards behaviors experimenting the interaction of multiple objects.

Here, we use a similar model and study different mechanisms for adaptively selecting alternative strategies to reach a toy, which were not studied in our previous work focused on evaluating the impact of hierarchical representations of sensorimotor spaces (Forestier and Oudeyer, 2016a). We hypothesize that not only do the type of

decision mechanism to select an action (matching law, greedy) influence the resulting behavior and match observations in infants as explained in Siegler’s models, but also the measure on which the decision is based, whether it is a competence measure, as in ASCM and SCADS, or an information-gain based measure such as learning progress.

To test this hypothesis, we designed an experimental setup with two phases. In the first one, the agents are autonomously exploring their environment through three sensory spaces (representing the hand, stick and toy), and can learn how to move their hand, how to grab an available stick, and how to reach a toy with either the hand or the stick. In a second phase, the agents use the same strategy selection procedure as in the first phase, but are now only exploring towards retrieving the toy, which mimics the incentive given by the mother to retrieve the toy in Siegler’s lab experiment (Chen et al., 2000). In Siegler’s experiment, several tools were available but only one allowed to grab the toy, and the tool strategy was defined as trying to use any of the tool to reach for the toy. We simplify this setup and we place only one tool in the environment so that the tool strategy only contains one type of actions and is easier to interpret. We measure the success rates to grab the toy and we study the evolution of the use of tool and hand strategies in this second phase depending on the mechanism of strategy selection, for individual agents.

Siegler (Siegler, 1996) suggests that the cognitive variability observed in infants could be essential to learning in childhood, and model it as matching law on the competence of the strategies. Our results suggests that an alternative mechanism, not proposed in Siegler’s model, could be at play in their tool-use experiment: the strategy selection could be based on a measure of learning progress instead of performance.

4.2.1 Methods

Environment

We simulate² a 2D robotic arm that can grasp a block or grasp a stick and use it to move the block. In each trial, the agent executes a motor trajectory and gets the associated sensory feedback. At the end of each trial, the arm and the stick are reset to their initial state, and the block is reset to a random location every 20 iterations. The next sections precisely describe the items of the environment and their interactions. See Fig.4.10 for an example state of the environment.

Robotic Arm The 2D robotic arm has 3 joints. Each joint can rotate from $-\pi$ to π (*rad*) around its resting position, which is seen by the agent as a standard interval $[-1, 1]$. The length of the 3 segments of the arm are 0.5, 0.3 and 0.2 so the length of the arm is 1 unit. The resting position of the arm is vertical with all joints at 0 *rad*

²Source code and notebooks available as a GitHub repository at <https://github.com/sebastien-forestier/ICDL2016>

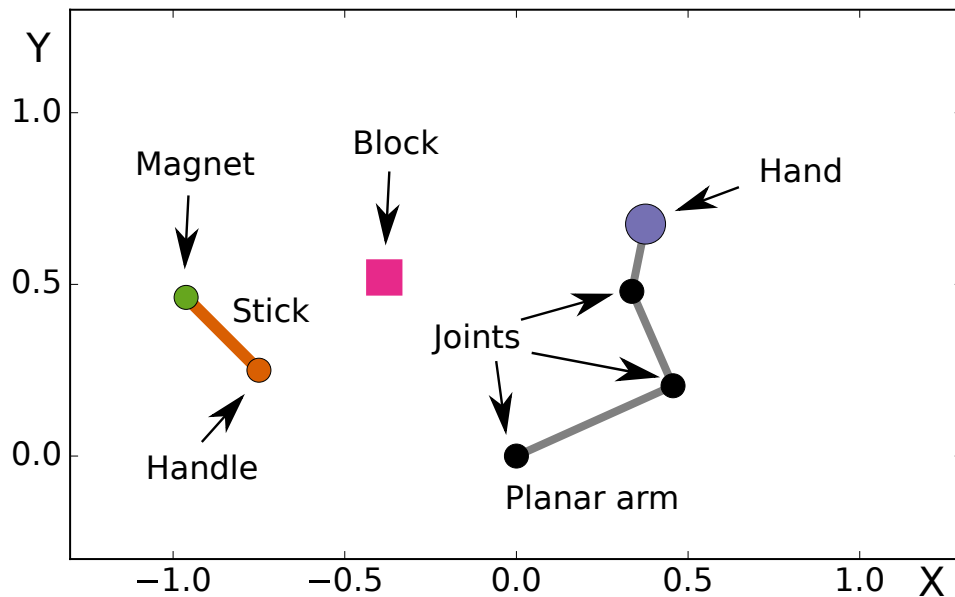


Figure 4.10: A state of the environment. The initial position of the arm is vertical so in this position the first and third joints are rotated to the right and the second joint to the left. The magnetic stick is at its initial position and is reset at each iteration. The block can be caught either by the magnetic side of the stick or directly by the hand as it is reachable here. The block is only reset every 20 iterations to a random position reachable by the hand.

and its base is fixed at position $(0, 0)$. A trajectory of the arm will be represented as a sequence of vectors in $[-1, 1]^3$.

Motor Control We use Dynamical Movement Primitives (Ijspeert et al., 2013) to control the arm’s movement. Each of the 3 arm’s degrees-of-freedom (DOF) is controlled by a DMP starting at the rest position of the joint. Each DMP is parameterized by one weight on each of 2 basis functions and one weight specifying the end position of the movement. The weights are bounded in the interval $[-1, 1]$ and allow each joint to fairly cover the interval $[-1, 1]$ during the movement. Each DMP outputs a series of 50 positions that represents a sampling of the trajectory of one joint during the movement. The arm’s movement is thus parameterized with 9 weights, represented by the motor space $M = [-1, 1]^9$.

Objects A stick and a toy (block) are available in the environment. The stick can be grasped by the handle side and can be used as a tool to catch the block. The stick has length 0.3 and is initially located at position $(-0.75, 0.25)$ as in Fig. 4.10.

If the hand reaches the block (within 0.2), we consider that the block is grasped until the end of this movement. Similarly, if the hand reaches the handle side of the stick (within 0.1), the stick is considered grasped and follows the hand’s position with the direction of the arm’s last segment until the end of this movement. If the magnetic side of the stick reaches the block (within 0.1), then the block follows the stick’s magnet.

Sensory Feedback At the beginning of each trial, the agent gets the context of the environment: the position of the block (*Context*, 2D). At the end of the movement, it gets sensory feedback from the following items in the environment. First, the trajectory of the hand is represented as its x and y positions at 3 time points: steps 17, 33, 50 during the movement of 50 steps (S_{Hand} , 6D). Similarly, the trajectory of the magnet of the stick is a 3-point sequence of x and y positions (S_{Stick} , 6D). It also gets the initial and final position of the block, and the minimal distance during the movement between the hand and the block, if the stick was not grasped, or between the magnet and the block, if the stick was grasped (S_{Block} , 5D). The total sensory space S has 17 dimensions.

Learning Agent

The problem settings for the learning agent is to explore its sensorimotor space and collect data so as to generate a diversity of effects in the three available sensory spaces, and to learn inverse models to be able to reproduce those effects. In this section we describe the hierarchical learning architecture.

Global Architecture of Sensorimotor Models The agent learns 4 sensorimotor models at the same time (see Fig. 4.11). Model 1 learns a mapping from the motor space M to S_{Hand} , model 2 from S_{Hand} to S_{Stick} , model 3 from S_{Hand} to S_{Block} and model 4 from S_{Stick} to S_{Block} . The block is the only item that can have a different initial position at the beginning of each iteration. We thus call contextual models the two models that have to take into account this context (models 3 and 4), and non-contextual models the two others (models 1 and 2). Those two types of models provide the inverse inference of a probable motor command m (in their motor space) to reach a given sensory goal s (in their sensory space), but their implementation is slightly different (see next sections).

In order to get interesting data to build its sensorimotor model, the agent performs Goal Babbling. It first chooses one of the three sensory spaces, and then self-generates a goal in the sensory space and tries to reach it. To generate those goals, different strategies have been studied (Baranes and Oudeyer, 2013). Here we use a random generation of goals for the exploration of spaces S_{Hand} and S_{Stick} (Random Goal Babbling), which was proven to be highly efficient in complex sensorimotor spaces

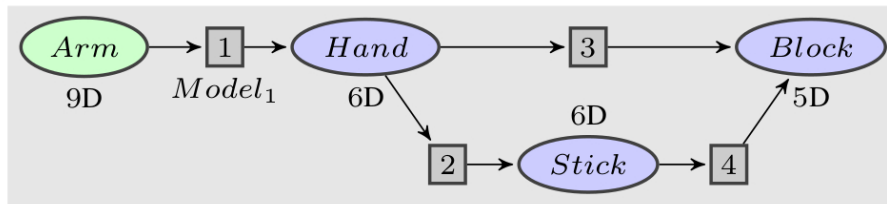


Figure 4.11: Architecture of models. The green circle is the motor space and the blue ones are sensory spaces. The gray squares are the 4 models.

(Rolf et al., 2010). For S_{Block} , we just define the goal as moving the block to the origin position $(0, 0)$.

If the goal is in S_{Block} , the agent also has to decide which method to use in order to try to retrieve the block: either the forward method, with model 3, or the tool method with model 4. In the other cases, if the goal is chosen in S_{Hand} or S_{Stick} , then model 2 or respectively 3 is used. Once the babbling model is chosen, it performs inverse inference and uses lower-level models to decide which motor command m will be experimented in the environment.

Finally, when motor parameters m have been tested in the environment and feedback s received, the mappings of models 1 and 2 are updated, and if the agent grasped the tool, then model 4 is updated, otherwise model 3 is updated. Also, a measure of success to reach the goal and of learning progress are computed and will be used to help choosing the space to explore. We use the Explauto autonomous exploration library (Moulin-Frier et al., 2014) to define those sensorimotor models and the learning progress measure.

Non-Contextual Models Each non-contextual model has a motor space (e.g. motor space of model 2 is S_{Hand}) and a sensory space (respectively S_{Stick}). They learn a mapping and provide the inverse inference of a probable motor command m (in its motor space) to reach a given sensory goal s (in its sensory space). They store new information of the form (m, s) with $m \in M$ being the experimented motor parameters and $s \in S_i$ the associated sensory feedback in their sensory space. They compute the inverse inference with the nearest neighbor algorithm: they look at the nearest neighbor in the database of a given s in the sensory space, and return its associated motor parameters. Model 1 also adds exploration noise (Gaussian with $\sigma = 0.01$) to explore new motor parameters.

Contextual Models The inverse inference is computed differently for contextual models (models 3 and 4). Whatever the position of the block (context), the agent tries to grasp it (with the hand for model 3 and with the tool for model 4) and to put it at the origin location $(0, 0)$. To do so, if the context is new (not within 0.05 of any previously seen context), then the agent chooses the motor command that

in the past led to the grasping of the block in the closest context. If the context is not new, then the model chooses the sensory point in the database with the smallest cost among the points that had a similar context (context within 0.05 of the current one), and a Gaussian noise ($\sigma = 0.01$) is added to the motor position. The cost of a sensory point s_{block} with context c is

$$cost(c, s_{block}) = D_{S_b}(traj, c) + D_{S_b}(origin, p_{final}) \quad (4.4)$$

where $D_{S_{block}}(traj, c)$ was the minimal distance between the hand (for model 3) or tool (model 4) and the toy during the trajectory. Also, $origin$ is the position $(0, 0)$ and p_{final} is the final position of the toy. Finally, D_{S_i} is a normalized distance in a sensory space S_i ,

$$D_{S_i}(s, s') = \frac{\|s - s'\|}{\max_{s_1, s_2} \|s_1 - s_2\|} \quad (4.5)$$

Active Space Babbling At each iteration, the architecture first has to choose the sensory space S_i to explore. This choice is probabilistic and proportional to the interest of each space (but with $\epsilon = 5\%$ of random choice). We call this procedure Active Space Babbling.

When space S_{Hand} is chosen to be explored, a random goal s_g (hand trajectory) is sampled and then sensorimotor model 1 is used to infer a motor command m to realize this hand trajectory. When space S_{Stick} is chosen, a random goal s_g (stick trajectory) is sampled and model 2 is used to infer a hand trajectory to make this stick trajectory (and model 1 used to realize the hand trajectory). When space S_{Block} is explored, then model 3 or 4 (hand or tool strategy) has to be chosen (see next section) to reach for the toy and the goal s_g is to catch the toy and put it at position $(0, 0)$.

We now define the learning progress and interest of a sensorimotor model mod that tries to reach the goal s_g (e.g. model 1 if S_{Hand} was chosen, or model 4 if S_{Block} and the stick were chosen). Once the motor command m is executed, the agent observes the current sensory feedback s in the chosen sensory space S_i . This outcome s might be very different from s_g as this goal can be unreachable, or because lower-level models are not mature enough for that goal. We define the progress $P(s_g)$ associated to the goal $s_g \in S_i$:

$$P(s_g) = D_{S_i}(s_g, s') - D_{S_i}(s_g, s) \quad (4.6)$$

where s_g and s are the current goal and reached sensory points, and s'_g and s' are the previous goal of the model mod that is the closest to s_g , and its associated reached sensory point. The progress of model mod is initialized at 0 and updated to follow the progress of its goals (with rate $n = 1000$):

$$P_{mod}(t) = \frac{n-1}{n} P_{mod}(t-1) + \frac{1}{n} P(s_g) \quad (4.7)$$

where t is the current iteration. The interest of model mod is its absolute progress, meaning that a negative progress is also interesting:

$$I_{mod}(t) = |P_{mod}(t)| \quad (4.8)$$

Now we define the interest of space S_{Hand} and S_{Stick} as the interest of models 1 and 2 respectively. The interest of space S_{Block} is the sum of the interest of models 3 and 4.

Choice of Method to Reach the Block When the agent has chosen to explore S_{Block} , and given a block position (context), it has to choose one of its two available methods to reach the block: the hand method (model 3) or the tool method (model 4). We define 4 conditions with different choices, based on two measures; competence and interest. The competence measure estimates for each method if the agent will be able to grasp the block. It is computed as follows: if the block was never grasped with the method, then it is -1 , otherwise it is the distance of the closest context where the block was grasped. The interest measure estimates the learning progress of each method to reach the current block position. If the context is strictly new, then the interest is the inverse distance of the closest context where the block was grasped (or 1 if there was no such context). If the context is not new, which means that the block was not grasped in the previous attempts, then the interest is computed as a derivative of the costs of the previous attempts for this context. If there were n previous attempts a_i , then the interest is

$$\left| \text{mean}_{\frac{n}{2}+1..n} [\text{cost}(a_i)] - \text{mean}_{1..\frac{n}{2}} [\text{cost}(a_i)] \right| \quad (4.9)$$

where the cost of an attempt is the one of Equation 4.4. Finally, for each of those two measures, we define two types of choice for both measures. The ϵ -greedy choice is a random choice with probability $\epsilon = 5\%$, and the choice of the highest with probability $(1 - \epsilon)$. In the matching law choice, the probability of choosing each method is proportional to the measure, but also with $\epsilon = 5\%$ probability of a random choice. This results in 4 possible conditions:

- GC: greedy on competence
- MC: matching law on competence
- GI: greedy on interest
- MI: matching law on interest

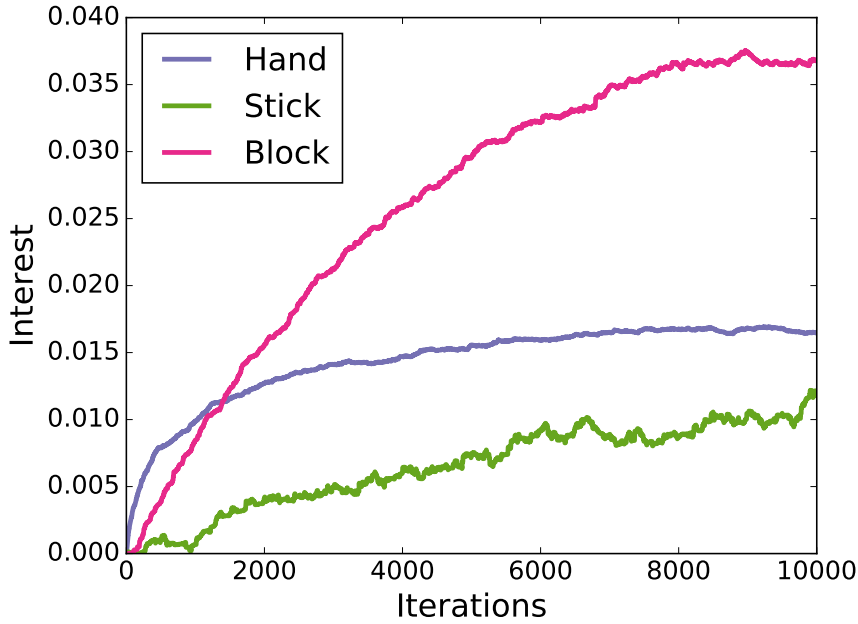


Figure 4.12: Evolution of the interest of spaces for one agent of condition MI during 10000 iterations of phase 1.

Experiments

The experimental procedure is composed of two phases. In phase 1, the agents are autonomously learning for 1000, 2000, 5000 or 10000 iterations where we reset the toy to a random position (but reachable directly with the hand) each 20 iterations. In phase 2, the agents are successively exposed to 3 new problems (or contexts) while they keep updating their sensorimotor models, for 200 iterations allowed per new problem. In those 3 problems, the toy is set at a location reachable with the tool but not reachable with the hand (problem A: $(-0.1, 1.2)$, B: $(0, 1.25)$, C: $(0.1, 1.2)$). Those locations are distinct enough so that given the solution to one of them, solving another one requires some exploration, but close enough so that the previous one helps. Finally, we simulate 100 independent trials for each condition.

4.2.2 Results

Figure 4.12 shows an example of the evolution of the interests to explore the sensory spaces during phase 1 for an agent of condition MI. After some iterations, the interest of S_{Block} becomes greater than the interest of S_{Hand} and S_{Stick} and thus is more often chosen to be explored.

Figure 4.13 shows in each condition and for each of the 3 problems of phase

2, the proportion of the 100 agents that succeeded to reach the toy, depending on experience (the number of iterations performed in phase 1, i.e. the number of sensorimotor experiments/movements already achieved by each agent). We see that in all conditions and for all problems, the success rate increases with experience. For instance, for problem A in condition MI, the success rate goes from 25% when agents have experimented 1000 iterations to 50% when they have experimented 10000. Also, for all conditions and experiences, the success rate increases from problem A to B and from problem B to C. For example, the success rate is 21% for problem A of condition MC at experience 1000 and it goes to 27% for problem B and 33% for problem C. Finally, the success rates of all problems in condition GC are smaller by 5 to 20% than the success rates of the three other conditions, and the success rates of condition MI are slightly higher than those of condition MC.

Figures 4.14 and 4.15 show 2D maps of the preference between the hand and tool strategies to reach the block depending on its 2D position (on a 100×100 grid), for one agent of experience 10000 iterations of each condition that succeeded to catch the block on the three problems. Also, the maps are computed at different times of phase 2 for each condition: at the beginning of phase 2 before problem A, after problem A, after problem B and after problem C. The preference is computed as the probability of choosing the hand strategy, and is reported on a two color scale. A completely blue region means that if the block is located in that region, then the corresponding agent would certainly (with probability 1) choose the hand strategy. This is almost the case in conditions GC and GI where the choice is ϵ -greedy with $\epsilon = 5\%$. Similarly, in green regions of those conditions, the choice is almost always for the tool strategy. However, a whiter region (in conditions MC and MI) means that the choice is more balanced, and in completely white regions the choice is equiprobable. It should be noted that the arm is located at position $(0, 0)$, has length 1, and can catch the block within 0.2 so it could theoretically reach the block within a circle of radius 1.2. However, in the 3 problems of phase 2, the block is unreachable directly with the hand. In those problems, the block is located at positions $(-0.1, 1.2)$, $(0, 1.25)$ and $(0.1, 1.2)$ (black dots).

In all conditions (from top to bottom) we can see modifications of the preference around those points across exposure to problems (from left to right), from a hand (blue) to a tool (green) preference. For instance, in condition GC (first row), before phase 2 (first column), this agent already preferred the tool. This is indeed possible because even if during phase 1 we reset the position of the block every 20 iterations to a random position reachable by the hand, this agent could have the time to move the block out-of-reach for the hand and then learn that it could catch it with the tool at that position. This is also part of the reason why success rate increases with experience in all conditions for problem A. Then, after the success to retrieve the toy in problem A (second column), the preference around problem A has changed in a small region around A, but towards the completely different choice: almost always choosing the tool strategy instead of always choosing the hand strategy. The results

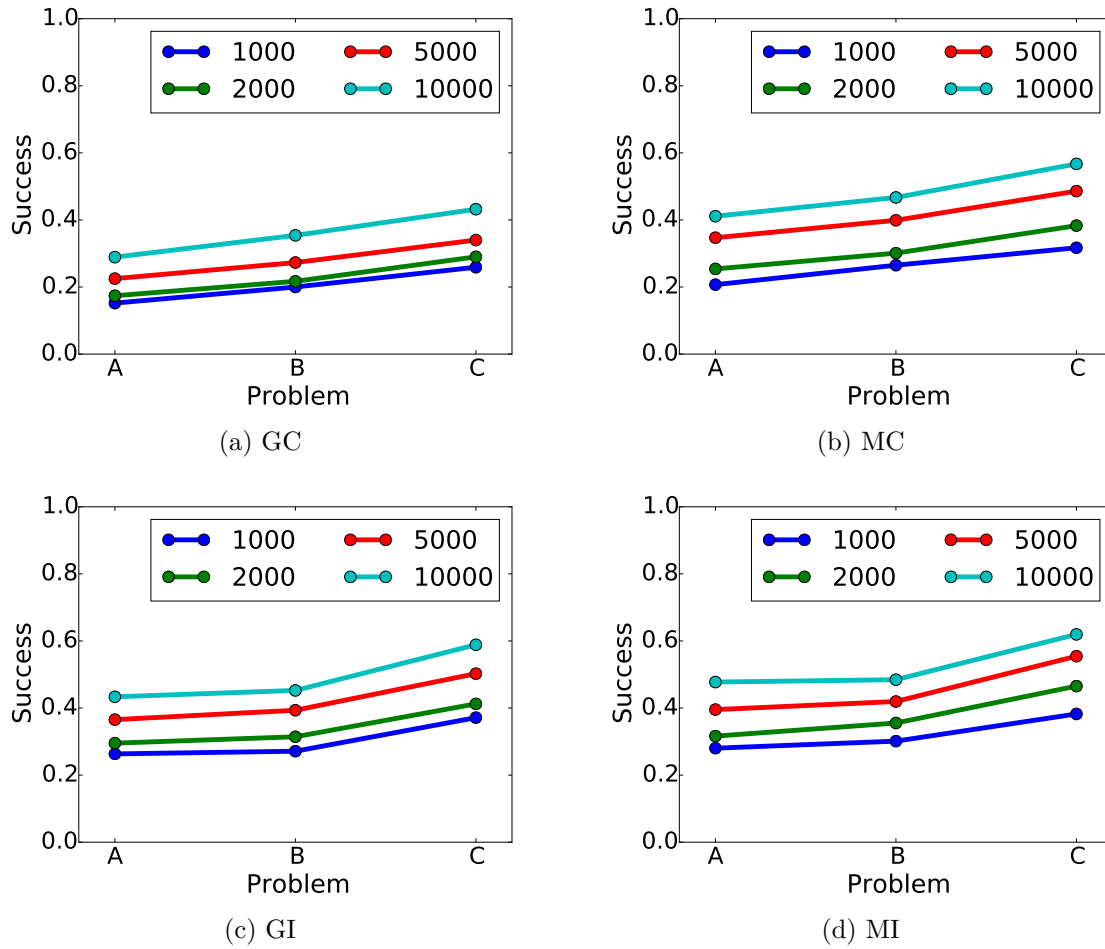


Figure 4.13: Proportion of 100 agents that succeeded to reach the toy in each of the 3 problems of phase 2, depending on condition and experience (the number of iterations experimented). Success rate increases with experience and with the problems encountered, and are better in conditions MC, GI and MI than GC.

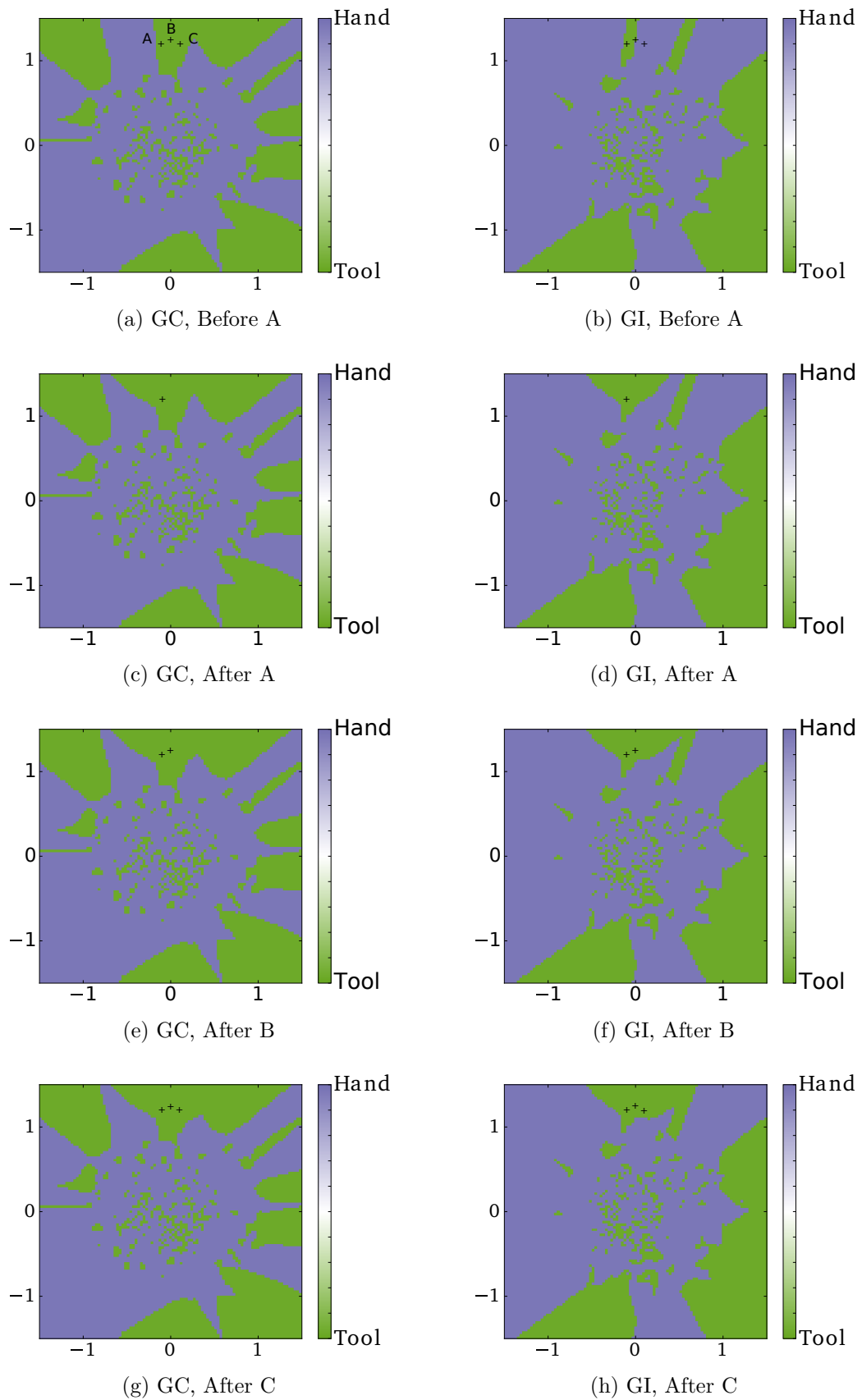


Figure 4.14: Strategy preference maps depending on condition and time point.

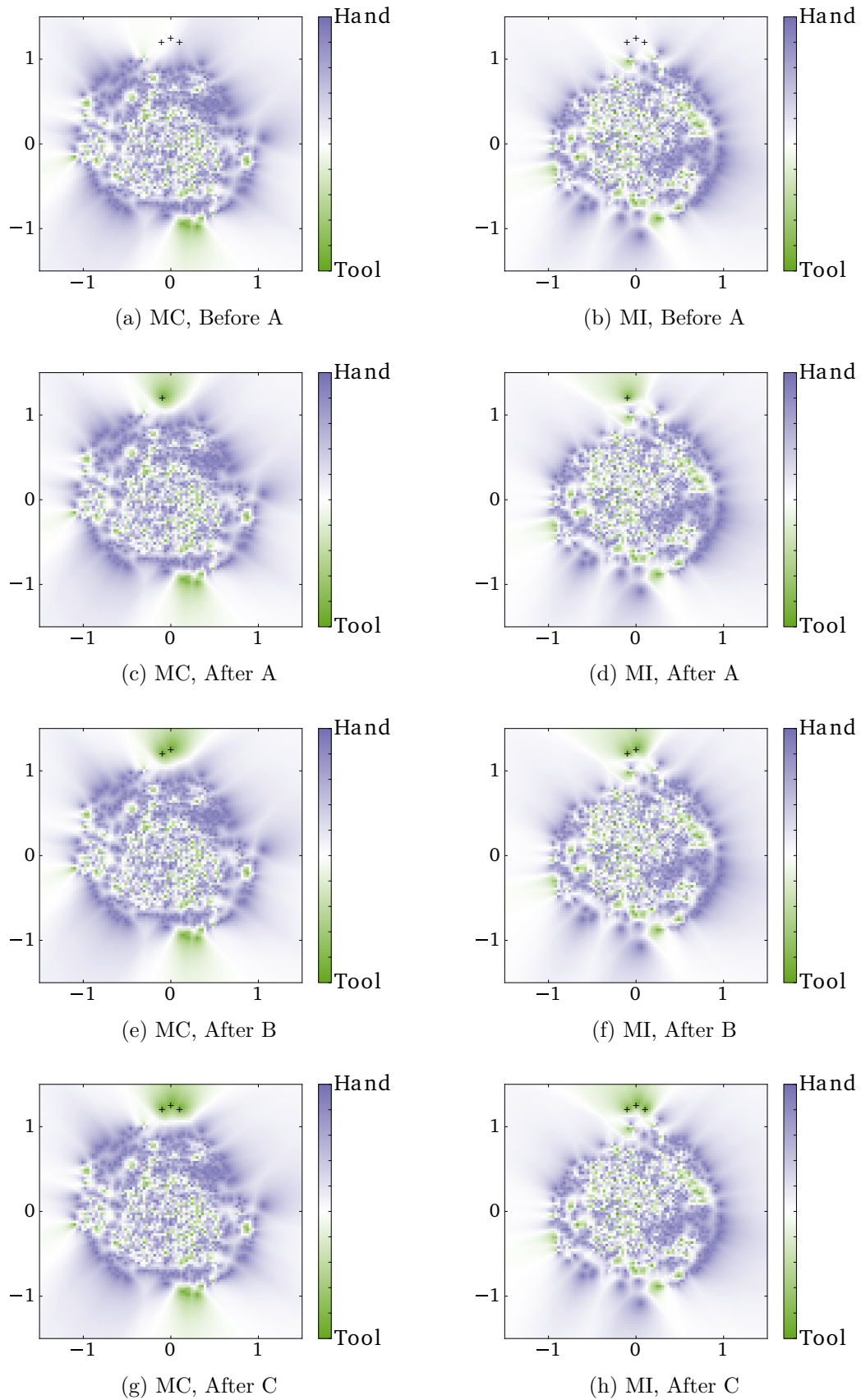


Figure 4.15: Strategy preference maps depending on condition and time point.

for the agent in condition GI are similar. However, the results for the agents of conditions MC and MI are different. In condition MC, the agent has no preference in problem A before phase 2, which means that for the first trial to retrieve the toy in problem A, it will choose the strategy randomly, and then the preference might change as the competence value depends on how far from the toy the strategy allowed to reach for. After problem A (second column), the preference changed in a large region around problem A, but this time the change is more gradual, with a high probability to choose the tool strategy only very close to A. The results for the agent in condition MI is similar, but here the preference before phase 2 was for the hand strategy (slightly: 60%, but for other agents it could have been for the tool strategy).

4.2.3 Discussion

We designed an experimental setup where an agent controlling a 2D robotic arm could learn two strategies to grab a toy depending on its position: grabbing the toy directly with the hand or first grab a tool to reach for the toy. Our setup is designed after the child experiment of Chen et al. (2000), and in particular their control condition where infants are encouraged to retrieve the out-of-reach toy. The agents have unified learning mechanisms for both free play exploration/learning of tool use from scratch modeling free play at home (phase 1) and for exposure to evaluation in lab sessions with an incentive to solve the task (phase 2). We defined two dimensions of strategy choice: the type of decision, with a matching law or a greedy choice, and the measure on which to make this choice: the performance of the strategies to retrieve the toy in its current position, or the progress made with each strategy to get the toy. The decision based on the performance measure means that the learner is interested in getting the toy, and the decision based on the learning progress means that the toy raises the curiosity of the learner about its affordances or relation with the hand and the stick.

The success rates in our setup are compatible with the ones of 1.5- and 2.5-year-olds in the experiment of Chen et al. (2000), where the success rates increase with experience and also across the successive problems. In their experiment, the toddlers in the control condition did use the other approaches than the tool strategy on more than half the trials after the first time they used the tool strategy in the lab experiment (84% of the trials for 1.5-year-olds, 48% for 2.5-year-olds). However, in the hint and modeling conditions, where the experimenter additionally suggested to use one of the available tools, or actively showed to the infant how to retrieve the toy with the tool, younger infants used other approaches in around 20% of the trials, and older ones in only 4%. Our setup is most similar to the control condition of Chen et al. (2000) as we did not model the hints and demonstrations given by the experimenter in the hint and modeling conditions. We observed that only our two conditions using a matching law, MC and MI display a concurrent use of the tool and hand strategies, with smooth evolution following new sensorimotor experience. The

behavior of agents in conditions MC and MI are thus compatible with the overlapping pattern observed with children in the control condition of Chen et al. (2000) where the mother just asked the child to get the toy.

Our results suggests that a strategy selection mechanism based on a measure of learning progress could be at play in the tool-use experiment of Chen et al. (2000). Also, condition MI could be more beneficial for learning in our setup than condition MC as success rates are slightly better in condition MI. An intrinsic motivation based on a matching law on performance could waste too many experimental trials on high-performing but not improving strategies, even with a novelty bias (that would expire irrespective of progress). On the contrary, a matching law on the monitored learning progress of each strategy could focus the training on low-performing but improving strategies and avoid wasting trials training high-performing but non-improving strategies. Also, a currently bad strategy could turn out later to be interesting for other related tasks and thus benefit from training while this is not considered in our model. On the other hand, an emphasis on learning progress might too often lead to the choice of an improving strategy that could turn to be sub-optimal or useless.

In our setup, at each iteration in phase 1 the agents have the choice to explore one of the three available sensory spaces. In phase 2, to model the incentive to get the toy given by the mother in the control condition of Chen et al. (2000), the agents were given the goal to retrieve the toy, and could choose either the hand or tool strategy to reach this goal. We thus assumed here that the goal of our agents in this test phase is the one decided by the experimenter, in order to focus on the strategy selection mechanism. However, as we discussed in chapter 3, it seems that in such tool-use experiments children are choosing their own goals that may often be different from the goal expected and made salient by the experimenter. We leave for future work the study of other possibilities to model the interaction between the encouragements to retrieve the toy (the extrinsic goal) and self-generated goals such as exploring the hand, the tool and the toy in many ways. To this end, it would be interesting to reproduce more closely the setup of Chen et al. (2000) as they placed several tools with different properties on the playground (sticks, rattles, hooks, etc), yielding on one hand different possible interactions with the toy and thus different potential learning progresses to control the toy with the tool, and on the other hand offering many potential alternative goals.

4.3 General Discussion

In the previous chapter, we found that in typical tool-use tasks infants could be choosing their own alternative goals and strategies different from the ones expected by the experimenter, which could interfere with the task results. Intrinsic motivations could have an important role in the development of tool use, however the details of

its mechanisms and its interaction with other forms of motivations remain unknown. In the present chapter, we used a computational modeling approach to investigate the mechanisms of tool-use development in infants. Indeed, the implementation of intrinsic motivations in artificial agents embodied in a robotic environment allows to evaluate the behaviors emerging from the interaction of the agent and its environment, and to study different hypotheses on the mechanisms of the learning of tool-use skills.

We extended previous models of intrinsically motivated learning to tool-use environments where the agents have to learn sensorimotor skills from scratch and discover that some objects can be used as tools to act on other objects. We first demonstrated that a modular environment representation corresponding to the tool-use objects is a determining factor for the emergence of structured behavioral phases in our simple tool-use setup. In our model, the active exploration of the environment with intrinsic motivations reinforced this emergence and is essential to efficiently learn in this tool-use environment. In a second experiment, we studied the evolution of strategy preferences in curiosity-driven artificial agents in a tool-use setup similar to one of the experiment of Chen et al. (2000). We showed that a choice of strategy based on learning progress could lead to the overlapping waves pattern observed by Chen et al. (2000), while it is not considered in Siegler's models of overlapping waves.

However, this modeling work has several shortcomings and limitations. First, young infants need to adapt to the maturation of their vision and to a developing body, while in those experiments we assume that the agent already has a good visual perception in that we provide it the position and trajectory of all objects in the scene. In a follow-up work, we study the possibility of learning a representation of the scene directly from pixels (see Appendix B). However, from a cognitive point of view regarding human tool-use learning, it is reasonable to suppose that the brain has sufficient knowledge about the concept of objects and their properties at the time of understanding object interactions and tool use, and it makes sense to build upon those representations in order to model tool-use learning (Lake et al., 2016). The modular representation we give to the learning agent thus seems natural as each sensory sub-space corresponds to the behavior of one object in the scene. Here, we further provide the structure of the sensorimotor hierarchy to the agent as a prior. The question of the autonomous learning of such a sensorimotor hierarchy is an important one but is left for future work. In chapter 8, we show that representing the objects in the tool-use environment as a hierarchy of means-end mappings is not a requirement for the emergence of tool-use behaviors, but exploring a modular set of mappings from primitive motor actions to each object is sufficient for developing tool-use behaviors. Similarly, in chapter 5, our agents learn to produce several words through vocal babbling and imitation, with no predefined hierarchy of sensorimotor mappings. A hierarchical representation together with planning and action sequencing abilities might still be necessary for the refinement of those skills, however the development of simple tool-use skills seems to be possible with this intrinsically motivated goal exploration.

In our work, the simulated robotic body is assumed constant over development. Previous work has shown that intrinsic motivations could be combined with maturational constraints to control the growth of complexity in motor development and thus scaffold learning (Baranes and Oudeyer, 2010b). A similar myelination process could be integrated in our framework with an initial limitation of the accuracy of motor control and of the capacity to discriminate objects, and a release of degrees of freedom following a proximo-distal direction. Another limitation of our setup is the fact that the robot and its interaction with tools and toys are all simulated. The dynamics of a real robotic arm and of its interaction with objects is certainly more complex than our simulation, with inaccuracies in the angular position of real motors, play in the mechanical parts, sensitivity to gravity and unpredictable collisions between objects, so that producing several times the same motor command do not yield the same results. In chapter 8, we show that an intrinsic motivations pushing the robot to explore objects for which it is making the most learning progress offers similar benefits in a real complex tool-use robotic setup compared to simulations.

Social guidance enables many pathways for learning in infancy, such as providing reinforcement or input for imitation and mimicry, or attracting the attention towards useful targets (see Background). Those mechanisms are of central importance for the sensorimotor development and in particular for tool use, but we do not address the question of its modeling in this work nor of the interplay between social guidance and self-exploration. For instance, the exploratory behaviors of babies are perturbed when social guidance is given. In the hint and modeling conditions of Chen et al. (2000), the experimenter respectively suggests to use the target tool or actively shows how to retrieve the toy with the tool. The strategic variability is much lower in those conditions, e.g. the 2.5-year-olds used other strategies than the tool one in only 4% of the trials after the first time they use it. We interpret this decrease in variability as the result of the incentive given by the experimenter, supposed to focus attention towards the target tool and to trigger the tool strategy. In our model, the hint condition could be integrated as a social bias to select the tool strategy. Also, the demonstration provided by the experimenter in the modeling condition could be added to the sensorimotor models as examples to reach the toy given the trajectory of the tool and the hand, the agent only having to find motor parameters to realize the hand trajectory (to solve the correspondence problem).

In the two experiments presented in this chapter, we implemented and compared several intrinsic motivations algorithms, with variants on the measure on which the preference is based or on the type of decision. In babies, several strategy selection mechanisms (e.g. based on competence or competence progress, with a greedy or exploratory selection) could be available across all situations, and they could switch between them or combine them depending on the estimated interest of exploration, the desire to actually get the toy, or social cues as the mother or experimenter incentive in Chen et al. (2000). A more sophisticated model can thus be envisioned, but to evaluate the matching between more complex models and actual infant data,

we would need a more in-depth confrontation of the models and the available data. Indeed, in our studies, in the end we only loosely reproduced some aspects of infant studies by accounting for abstract structures in child development such as the high-level structure of the evolution of behaviors across sensorimotor development and in tool-use sessions. In order to discriminate more complete models, we would in any manner need more accurate and high-density data in sensorimotor interactions. For instance, the success rates in trials as reported by Chen et al. (2000) would not be enough to discriminate the real-time mechanisms of several possible models. Recent infant studies have made efforts in this direction. In DiMercurio et al. (2018), fine-grained measurements of the movements of each limb of babies are recorded in short sessions across their first 2 months. In Bambach et al. (2018), head-mounted eye trackers allow to record the scene as viewed through the eyes of a baby in real time. The high-density data gathered with those kinds of setups could fuel future modeling efforts.

Although mechanisms such as sequential learning, causal inference or action observation are known to be important in child development, we suggest that intrinsic motivations should be considered as one of them, but has comparatively little been studied so far. The understanding and modeling of intrinsically-motivated exploration in child experiments and of its impact in child development still offer many challenges: How to model intrinsically-motivated exploration behaviors ? Are children using several motivational mechanisms at the same time and depending on the learning context ? What features of the environment could be part of a context that influence exploration and learning ? How does each type of social interaction impact intrinsic motivations processes ? How do the intrinsic motivations interplay with extrinsic motivations and basic drives in children and how could their interaction be modeled in robots ? Do the properties of intrinsic motivations change with development along a maturational clock ?

In the next chapter, we study the joint development of speech and tool use by intrinsically motivated agents embodied in a naturalistic simulated learning scenario, where the learning of speech production is grounded in the physical environment.

Chapter 5

A Unified Model of Speech and Tool-Use Early Development

Summary

Several studies hypothesize a strong interdependence between speech and tool-use development in the first two years of life. To help understand the underlying mechanisms, we present the first robotic model learning both speech and tool use from scratch. It focuses on the role of one important form of body babbling where exploration is directed towards self-generated goals in free play, combined with imitation learning of a contingent caregiver. This model does not assume capabilities for complex action sequencing and combinatorial planning which are often considered necessary for tool use. Yet, we show that the mechanisms in this model allow a learner to progressively discover how to grab objects with the hand, how to use objects as tools to reach further objects, how to produce vocal sounds, and how to leverage these vocal sounds to use a caregiver as a social tool to retrieve objects. Also, the discovery that certain sounds can be used as a social tool further guides vocal learning. This model predicts that the grounded exploration of objects in a social interaction scenario should accelerate infant vocal learning of sounds to name these objects as a result of a goal-directed exploration of objects (Forestier and Oudeyer, 2017).

Several studies hypothesize that there might be a strong interdependence between speech and tool-use development in the first two years of life (Gibson et al., 1994; Greenfield, 1991). Tool use and language seems to require similar information processing capabilities allowing the production and perception of sequential combinations of increasing complexity, from reaching to spoon self-feeding and from words to stories. In addition to displaying similar compositional properties, speech and tool use might share some neural correlates involving Broca’s area (Higuchi et al., 2009). Those common neural correlates could have an evolutionary origin in the hominid lineage, where a selection pressure for complex tool use, language and social behaviors might have together driven the increase in brain planning capabilities (Morgan et al., 2015), see Background. In particular, the development of tool-use precursors follows several overlapping phases of behaviors: 1) body babbling, where babies learn to control their body parts, 2) interacting with a single object, and 3) exploring object-object interactions (Guerin et al., 2013). From pointing movements to the control of a rake, new representations and physical understanding are developed to allow the planning of tool-use actions composed of combinations of more simple actions, e.g. grasping the rake. During the same period, infants progressively learn how to efficiently use their vocal tract, comprising many complex actuators from the larynx to the lips. At birth, they produce immature protophones like squeals, growls or quasi-vowels, and by the end of their first year they are able to produce the speech-like syllables of their native language (Oller, 2000). Those syllables then form words which become the basis of syntactic combinations essential to language expressiveness. Infants do not only explore tool use and vocalizations by themselves, driven by intrinsic motivations (Moulin-Frier et al., 2013), but also spend a great part of their time interacting with their parents and other social peers, where imitation is thought to be one of the important developmental pathways (Meltzoff and Warhol, 1999). For instance, infants imitate the vowels produced by an adult speaker by 6 month of age (Kuhl, 2004), and 1.5-year olds imitate demonstrations of a rake-like tool function to retrieve an out-of-reach toy (Chen et al., 2000).

In order to investigate hypotheses about the joint ontogenetic development of speech and tool use, we seek to build an embodied model of tool use and speech learning. Existing robotic models of tool use showed first insights into how relations between tools and other objects could be learned from grounded experimentation. In Stoytchev (2005), a robotic arm focused on learning rake-like tool affordances from the exploration of already implemented stereotyped arm behaviors. In Tikhanoff et al. (2013), the iCub robot was given its arm’s forward model and inverse optimization methods which led to stereotyped grasping. In the previous chapter, we designed a series of robotic models considering the learning of tool use from scratch, without any kind of pre-programmed reaching skills, through the intrinsically-motivated exploration of a tool-use environment by embodied agents (Forestier and Oudeyer, 2016a,c). We studied the developmental progression between phases of behaviors with objects and tools and the evolution of their strategies to reach goals, depending

on the implementation of goal-directed exploration and on the type of representation of the environment. We have shown that those models could reproduce aspects of the observation of infant development, in terms of developmental trajectories and strategy choice dynamics. In chapters 6, 7 and 8, we present, formalize and evaluate a related intrinsically motivated algorithmic architecture called Model Babbling (Forestier and Oudeyer, 2016b), with the difference that the sensorimotor representation has no pre-defined hierarchical structure, only a modular representation based on the objects of the environment. We show that the intrinsic motivations operating on this modular representation allows the development of tool-use skills, in simulated and in real robotic environments.

Recent computational models of vocal development make use of a simulated vocal synthesizer that the learning agent must control in order to produce vocalizations, with human sounds as targets to be imitated (Philippsen et al., 2014; Warlaumont et al., 2013). The DIVA model (Guenther, 2006) is a neural network simulating the cortical interactions producing syllables, that provides an account of different production phenomena, such as co-articulations. In a neural network model of motor prespeech production with self-organizing maps (Warlaumont et al., 2013), a reinforcement based on the similarity of the model's output sounds with a given set of vowels biases the post-learning model's babbling sounds towards that reinforced set of vowels. The Elija model (Howard and Messum, 2011) uses an articulatory synthesizer to produce sounds, and gets a reward for the exploration of its vocal outputs. The model also interacts with a caregiver that imitates its sounds like a mother would do: either mimicking the infant's sounds or providing an intermediate sound between the infant's one and the adult one. The agent manages to learn object names by trying to reproduce caregiver's utterances. The agent of Philippsen et al. (2014) uses a recurrent neural network to learn the forward and inverse model of the VocalTractLab speech synthesizer. Their learning algorithm allows an efficient use of human supervision in the form of few examples of consonant-vowel sequences to be imitated. In Moulin-Frier et al. (2013), the agent chooses the strategy that shows the best competence progress: either autonomously training to reach phonetic goals, or trying to imitate human sounds. They show that the intrinsic motivation for learning progress self-organizes coherent infant-like developmental sequences. Those models of language acquisition study several developmental pathways to the learning of forward and inverse models of a simulated vocal tract, from autonomous exploration to human sounds imitation. However, agents are not situated into a physical environment where vocalizations have a meaning related to physical objects that can be interacted with.

Several works study joint action and language learning, but give an advanced knowledge of the linguistic interaction protocol to the learning agent who has to associate predefined actions or objects to predefined labels and learn the semantic compositionality (Billard, 1999; Cangelosi et al., 2010; Roy, 2002). In Sugita and Tani (2005), a wheeled robot with an arm learns to associate lexical symbols to behavioral categories through supervised learning (point, push, or hit the red, blue, green, left,

center, or right object). Dominey et al. (2009) designed a robot-human interaction scenario where the HRP-2 humanoid robot is able to understand the meaning of new linguistic instructions (such as "Give me X") by grounding them with preexisting motor skills. In Massera et al. (2010), a simulated robotic arm controlled by a neural network manipulates objects on a table, with linguistic instructions as input in the form of three values that represent the type of behavior that the robot should exhibit. In Tikhonoff et al. (2010), a simulated iCub is given a speech understanding module, a vision module, and a dataset of speech instructions, visual objects and corresponding expected actions, and has to learn the actions to perform depending on the instruction and the object in the scene. To our knowledge, there is no robotic model able to learn to produce words that have a meaning in a physical environment starting from scratch through the exploration of a vocal synthesizer and possibly in interaction with a peer.

In this chapter we describe a first model that jointly considers the early development of both tool use and speech. Such a model could allow the investigation of hypotheses about the mechanisms underlying the observed links between tool use and speech development. We build upon the Model Babbling architecture (Forestier and Oudeyer, 2016b) that leverages several fundamental ideas. First, goal babbling is a powerful form of exploration to produce a diversity of effects by self-generating goals in a task space (Baranes and Oudeyer, 2013). Second, the possible movements of each object define task spaces in which to choose goals, and the different task spaces form an object-based representation that facilitates prediction and generalization, as explained by Chang et al. (2016). A novel insight of this series of architectures was that early development of tool use could take place without requiring combinatorial sequencing and action planning mechanisms: modular goal babbling in itself allowed the emergence of nested tool-use behaviors. Here we extend this architecture so that the agent can imitate caregiver's sounds in addition to autonomously exploring. We hypothesize that these same algorithmic ingredients allow a unified development of speech and tool use from scratch. To study the joint development of speech and tool use, we situate our learning agent in a simulated environment where a vocal tract and a robotic arm are to be explored with the help of a caregiver in a naturalistic learning scenario. The environment is composed of three toys, one stick that can be used as a tool to move toys, and a caregiver moving around. The caregiver helps in two ways. If the agent touches a toy, the caregiver produces this toy's name, but otherwise produces a distractor word as if it was talking to another adult. If the agent produces a sound close to a toy's name, the caregiver moves this toy within agent reach.

Our results show that Model Babbling allows agents to learn how to 1) use the robotic arm to grab a toy or a stick, 2) use the stick as a tool to get a toy, 3) learn to produce toy names with the vocal tract, 4) use these vocal skills to get the caregiver to bring a specific toy within reach, and 5) choose the most relevant of those strategies to retrieve a toy that can be out-of-reach. In this experiment, the grounded exploration

of toys accelerates the learning of the production of accurate sounds for toy names once the caregiver is able to recognize them and react by bringing them within reach, compared to distractor sounds without any meaning in the environment. This is a first model allowing the study of the early development of tool use and speech in a unified framework.

5.1 Methods: a Grounded Play Scenario with a Caregiver

5.1.1 Learning Environment

The learning environment¹ is composed of a simulated 2D robotic arm and a simulated vocal tract that the agent controls to interact with a caregiver and toys. In each trial, the agent observes the current environmental state and then executes a motor trajectory, either corresponding to moving the motors of the arm or of the vocal tract, and gets the associated sensory feedback composed of the trajectory of each object and the sound produced by the agent or the caregiver (see Fig.5.1).

Simulated Robotic Arm

The simulated 2D robotic arm has 3 joints, with its base fixed at position $[0, 0]$. Each joint rotates from $-\pi$ rad to π rad and the 3 segments of the arm have length 0.25, 0.15 and 0.1, so the arm has length 0.5. The framework of Dynamical Movement Primitives (Ijspeert et al., 2013) is used to generate smooth joint trajectories from motor parameters. Each of the 3 joints is controlled by a DMP starting at the rest position of the joint (position 0) and parameterized by 7 weights: one weight on each of 6 basis functions and one weight representing the end position of the joint trajectory. To sum up, the agent provides a set of 21 trajectory parameters which are translated through DMPs to a set of smooth 50-steps trajectories for the arm's joints which gives a smooth 2D trajectory to the robotic hand.

Tool and Toys

In the environment of the robotic arm, 3 toys can be grasped with the hand or with the help of a stick. The stick has length 0.25 and is considered grasped as soon as the hand reaches the handle side (orange) within a distance of 0.1. At the end of the movement the stick is dropped and stays at its current position while the arm is reset to its rest position for the next iteration. The toys are reset to a random location

¹Source code and notebooks available as a GitHub repository at <https://github.com/sebastien-forestier/CogSci2017>

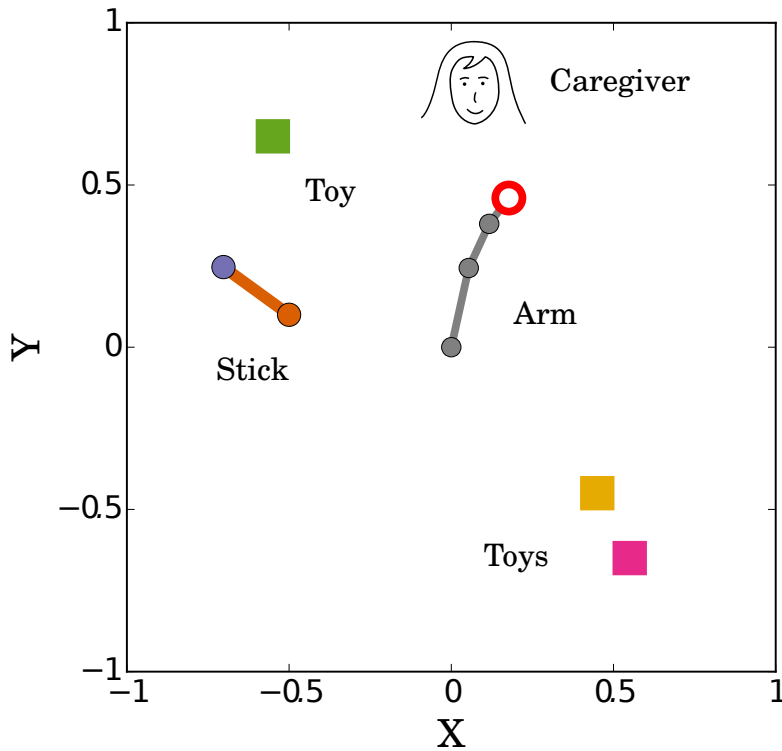


Figure 5.1: Agent's 3 DOF arm, controlled with 21 parameters, grabs toys with its hand, or uses the stick to reach toys. Caregiver brings a toy within reach if the agent says its name. When agent touches a toy, caregiver says toy's name.

every 20 iterations, at a distance between 0 and 1 from the center so possibly at an unreachable position.

Simulated Vocal tract

A vocal tract is simulated through the DIVA model (Guenther, 2006) and allows the production of different sounds that we can classify into vowels. In the DIVA model, a set of parameters defines a vocal tract contour where each represents one component of a Principal Component Analysis of midsagittal MRI vocal tract profiles (see Fig.5.1b), from the jaw and tongue to the lips position. Here we use only the first 7 articulatory parameters, controlling most of the vocal tract shape's variability. From a vocal tract contour defined by a set of parameters, the DIVA software computes the corresponding sound and outputs its first 2 formants, which are often considered to give enough information to distinguish common English vowels. The DMP framework generates smooth trajectories of vocal parameters, as described above for arm parameters, to allow the simulated vocal tract to produce simple words composed of several vowels. Each of the 7 articulators is controlled by a DMP parameterized by 4 weights: the

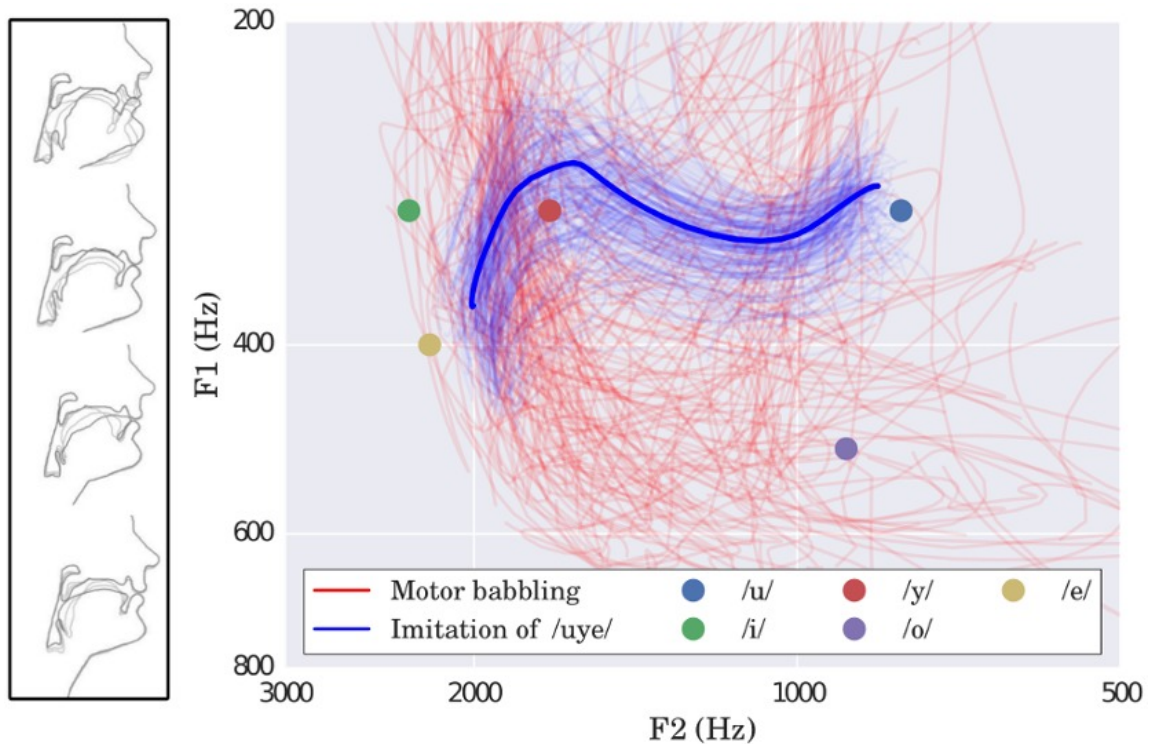


Figure 5.2: Agent’s vocal environment representing sounds as trajectories in the two first formants space. Agent’s simulated vocal tract produces sounds given 28 parameters. When agent touches a toy, caregiver says toy’s name. Some sounds corresponding to random parameters are plotted in red, and some sounds produced when imitating caregiver’s /uye/ word in blue (best imitation in bold, error 0.3).

starting and end position of the parameter trajectory, and weights on 2 basis functions. Given a set of 28 trajectory parameters provided by a learning agent, the DMPs output a set of smooth 50-steps trajectories for the 7 articulators that we use in the DIVA model, which through the DIVA software generates a smooth trajectory of the first two formants (called $F1$ and $F2$).

Sounds: from Vowels to Words

The simulated vocal tract controlled through DMPs has the potential to produce words composed of a sequence of 3 vowels in the set $\{/o/, /u/, /i/, /e/, /y/\}$. See Fig. 5.1 (b), ”Motor babbling” condition, for an example of 200 trajectories corresponding to random sets of 28 parameters. We define a set of 6 words that the caregiver produces perfectly: $\{/yeo/, /euy/, /iuo/, /uye/, /eou/, /oey/\}$. A sound trajectory produced by the vocal tract is recognized if its distance to the perfect word is lower than 0.4.

Caregiver's guidance

A simulated caregiver is given two roles to help the learning agent. First, at the beginning of the experiment, the caregiver chooses randomly a label for each toy from the set of 6 words. When the agent touches a particular toy with its hand, the caregiver then produces the sound trajectory corresponding to the label of this toy. If the agent does not touch any toy with the arm, the caregiver produces one of the distractor sounds, as if she was talking to another adult. Second, if the agent produces a sound trajectory recognized by the caregiver as the label of a toy, the caregiver moves the corresponding toy in between herself and the agent so that it becomes reachable by the agent with the hand. The caregiver is reset to a random position at each iteration.

Sensory Feedback

Before choosing a motor command, the agent receives the state of the environment (or context) as the 2D position of the caregiver, the stick and the 3 toys (so 10D). At the end of the movement, the agent receives a sensory feedback s in the sensory space S (60D), from the different objects in the environment. First, the trajectory of the hand is represented as its x and y positions at 5 time points: steps 1, 13, 25, 38, 50 of the 50-steps trajectory (S_{Hand} , 10D). Similarly, the trajectories of the stick and the 3 toys during the movement are represented in 10 dimensional sensory spaces (S_{Stick} , S_{Toy_1} , S_{Toy_2} , S_{Toy_3} , 10D each). Sound, either produced by the agent or by the caregiver, is represented by the position of the first two formants at 5 time points (S_{Sound} , 10D).

5.1.2 Unified Modular Learning Architecture

The goal of a learning agent is to use its robotic arm and vocal tract to discover a diversity of sensory effects, and collect data to learn repertoires of skills in the form of inverse models allowing to reproduce these effects. Consequently, the agent is not given a priori a single target task to be solved, but a modular object-based representation of task spaces. The agent learns a set of sensorimotor models mapping a motor space to one particular sensory space (see Fig. 5.3). For instance, model 1 learns to move the hand from arm parameters, model 2 learns to move the stick, model 3, 4, and 5 learn to move one of the toys, and model 6 how to produce sounds with the arm, which will be possible by touching one of the toys with the hand so that the caregiver produces the corresponding label. Controlling vocal tract, model 7, 8 and 9 learn to move one of the toys by involving caregiver's help, and model 10 learns to produce diverse sounds autonomously.

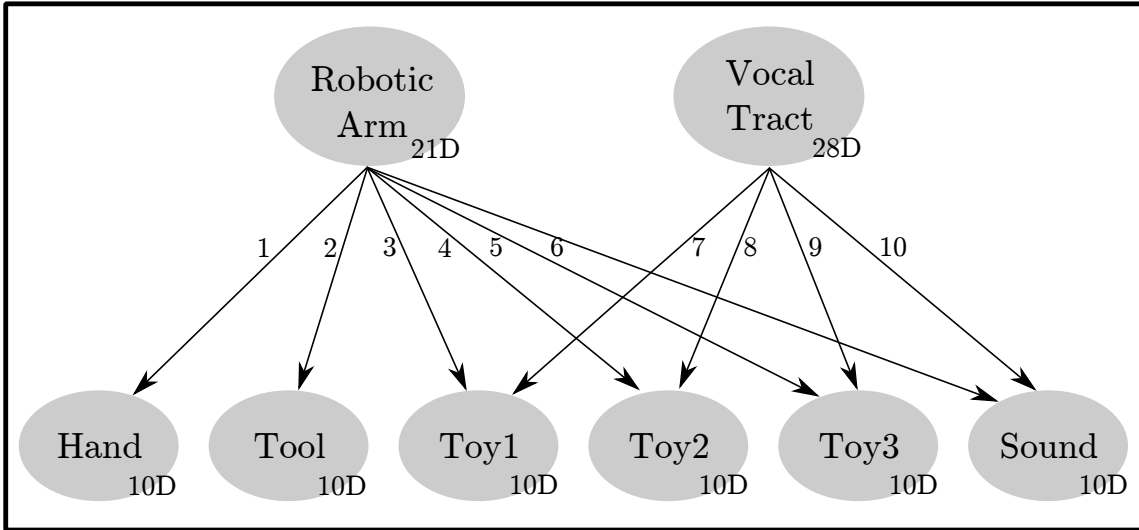


Figure 5.3: Learning Architecture. Agent controls 2 motor spaces and receives sensory feedback about 6 objects. Each arrow represents one of the 10 sensorimotor models learned.

Exploration through Model Babbling

In order to actively explore and learn the 10 sensorimotor models from experimentation with the environment, learning agents use the Model Babbling architecture developed in Forestier and Oudeyer (2016b) that we extend to handle the 2 motor spaces: the robotic arm and the vocal tract. First, the agent performs some random exploration of motor spaces, 500 with the robotic arm and 500 with the vocal tract, to get an initial sampling of those spaces. Then, at each iteration, the learning agent first chooses to train one of the 10 models, chosen randomly (e.g. from the robotic arm to the hand sensory space). A particular goal is then randomly chosen in the sensory space corresponding to the chosen model (e.g. a particular 2D trajectory of the hand). The agent then uses the corresponding inverse model to infer a motor command in the corresponding motor space (e.g. arm parameters) to reach the goal. Exploration happens in goal choice and in the new motor parameters that inverse models infer with generalization mechanisms and adding exploration noise.

Imitation of Sounds

When the agent is choosing to train to produce sounds with its vocal tract (model 10), instead of always choosing random goals, it does this for half of the iterations (chosen randomly), and the other iterations are focused on trying to imitate the caregiver, by randomly choosing one of the sounds previously produced by the caregiver as a goal.

Forward and Inverse Models

Each sensorimotor model provides a forward model and an inverse model, with the same implementation as in Forestier and Oudeyer (2016b). The forward model predicts which sensory trajectory would be observed given the current context and a motor command to execute. The inverse model infers a motor command that could reach a desired goal given the current context. When a motor command m is executed (either 21 parameters for the robotic arm or 28 for the vocal tract) in a context c and a sensory feedback s is received in S , all the sensorimotor models that share the same motor space are updated. New information comes as a tuple (m, c_i, s_i) with s_i being a subset of s variables corresponding to the respective sensory space, and c_i being the subset of c relevant for this sensorimotor model. The relevant context for models 1 and 10 is empty, which means that hand trajectories and vocal sounds produced by the agent do not depend on the current position of other objects. The context for model 2 is the position of the stick, and for models 3, 4, and 5, the position of the stick and of the corresponding toy. For model 6, all toys are relevant, and for models 7, 8 and 9, the caregiver and the toy is useful. Given a database of (m, c_i, s_i) experiments, an inverse model infers a probable motor command m to reach a goal s_g in a context c_i by looking for the nearest neighbor s_{NN} in S_i of s_g and retrieving the associated motor parameters m_{NN} that were used to reach s_{NN} . It then outputs m_{NN} plus Gaussian noise ($\sigma = 0.05$) to explore new parameters.

5.2 Results: Learning Tools and Words

We ran 500 independent trials of 80000 iterations (or robot experiment) each. We measured how agents learned to move objects by giving them new goals in new contexts, and we analyzed the accuracy of the learned vocalizations.

5.2.1 Competence to Reach Toys

After 80000 iterations of training, we measured the performance of each agent to retrieve a toy depending on its current position with its preferred method: with the hand, with the stick used as a tool or involving caregiver's help. Fig. 5.4 shows the mean competence of all agents to retrieve toys depending on the current position of the toys. The competence error to retrieve a toy is measured by the distance between a goal trajectory given to the agent, where the toy is moved towards the center, and the actual trajectory that the agent succeeds to give to the toy. The agent chooses the strategy expected by its inverse models to best reach the goal trajectory for the toy given the current context (position of the stick, toys and caregiver) and its past experience of 80000 iterations.

In most toy locations, the normalized competence of learning agents is significantly better (46% on average) than the normalized competence of a random agent (0%).

Our learning architecture thus allows to successfully reach new goals in multiple sensory spaces with multiple available strategies. Local variations reflect differences in strategy preferences and performances. For instance, where the hand cannot reach for the toy anymore, the agent still thinks this is a good strategy as it worked in a similar context (before the limit), but the hand strategy leads there to a bad performance. More training would refine the inverse models and the choice of strategy.

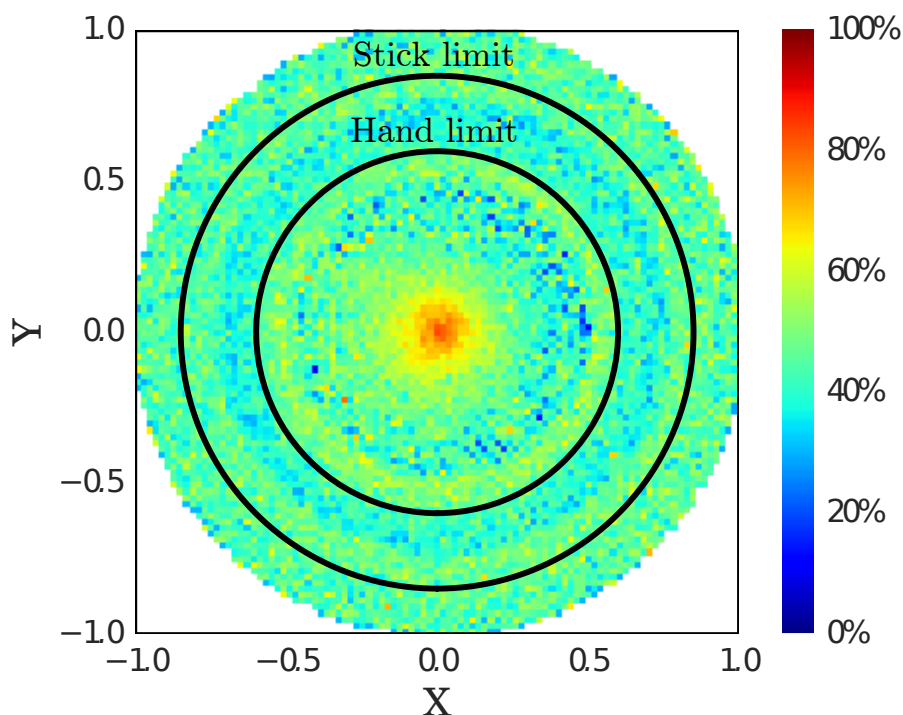


Figure 5.4: Competence after 80000 iterations. 0% means that competence to retrieve a toy there is as bad as with random agents, 100% says that agents perfectly retrieve a toy there.

5.2.2 Three Strategies to Reach Toys

Fig. 5.5 shows the preference for the hand, tool and vocal strategies to retrieve a toy depending on the distance of the toy. In the center region, where agents can retrieve toys with all three strategies, agents choose most often the hand strategy (around 65% of the trials) and less the other two (around 15% to 20% each). In the second region, unreachable with the hand, this strategy is still used around 50% of the trials, and the two other between 20% and 30%. In the last region where the only useful strategy is to say the name of the toy so that the caregiver brings it closer, the vocal strategy is used more often: at distance 1 from center, it is used in 49% of trials, hand strategy in 38%, and tool strategy in 13%.

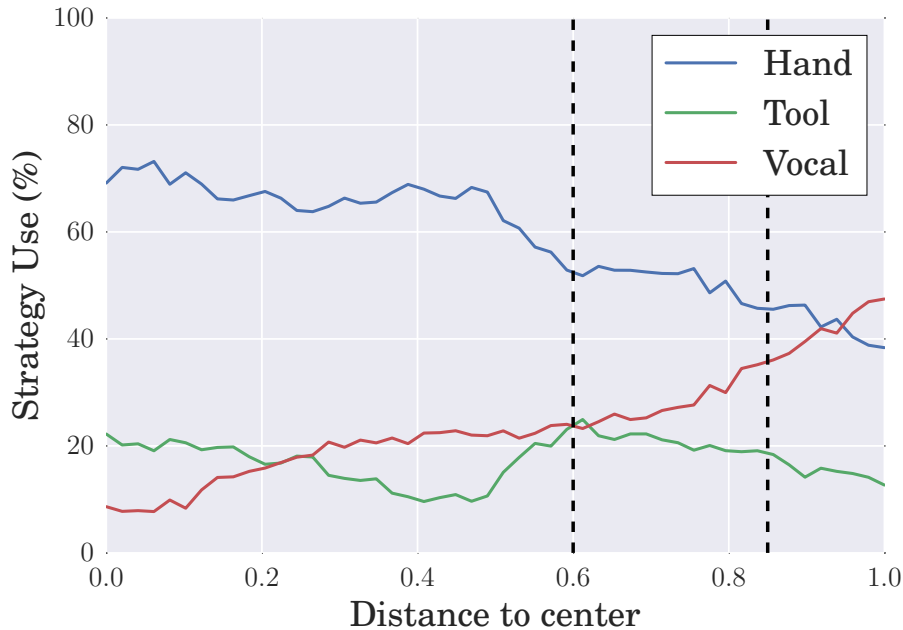


Figure 5.5: Strategy preferences depending on the distance of the toy. The two vertical bars shows the hand and stick limits.

5.2.3 Vocal Learning with Caregiver’s Feedback

The agents learn to produce vocalizations both with goal babbling and imitation of the caregivers’ sounds. For each agent, three of caregiver’s sounds (randomly selected) are toy names and the three others are distractors: sounds that have no special meaning for the agent. We measure errors to reproduce caregiver’s sounds as the distance between the sound trajectory produced by the caregiver and the best imitation of the agent. We group the results into two categories: errors of sounds that serve as toy names and as distractors. From the 500 runs we could retrieve error data for 1482 toy names and 1482 distractors. Fig. 5.6 shows the distribution of errors after 80000 iterations. First, 88% of sounds have an error lower than 0.4, and thus are successful imitations. Second, the median error for toy names is 0.23 and for distractors is 0.30. Imitations of toy names are more accurate than of distractors: a Mann-Whitney U test gives $p < 10^{-72}$. Errors distribution above 0.4 is similar for the two categories, but few toy name sounds have an error just below 0.4 compared to distractors: their distribution is shifted towards smaller errors.

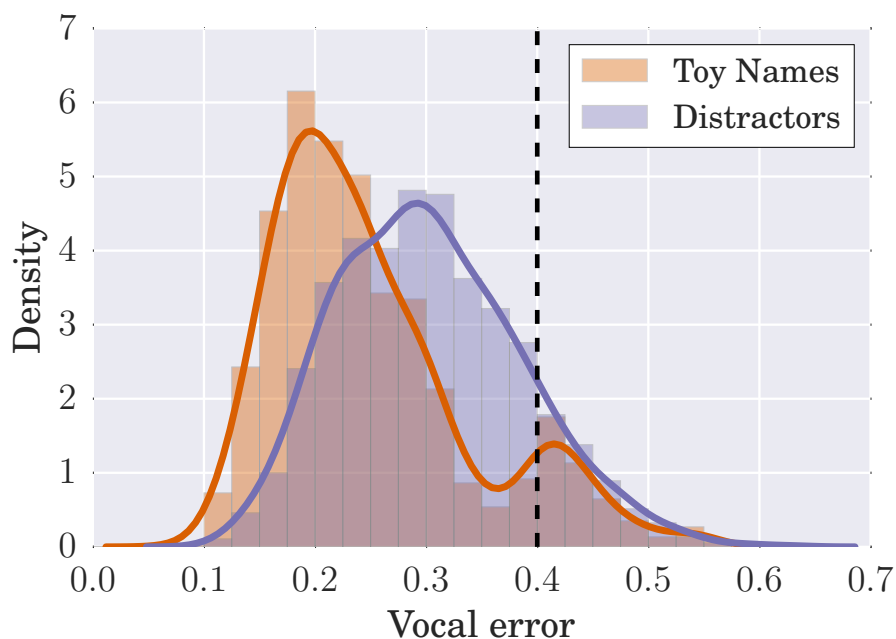


Figure 5.6: Distribution of accuracy of imitations of caregivers' sounds after 80000 iterations. Below 0.4 vocal error, sounds are recognized as imitations by the caregiver. Imitations of toy names are more accurate than imitations of distractors.

5.3 Discussion

The results of this study show that agents learned to 1) use the robotic arm to grab a toy or a stick, 2) use the stick as a tool to get a toy, 3) produce toy names with the vocal tract, 4) use these vocal skills to get the caregiver to bring a specific toy within reach, and 5) choose the most relevant of those strategies to retrieve a toy, for instance preferring to use caregiver's help when the toy is out-of-reach. Interestingly, learning the production of accurate sounds for toy names was faster than for distractor sounds. Indeed, once the agent succeeded to produce a toy's name close enough so that the caregiver could recognize it and react by pushing the toy towards the agent, the agent started to learn how to use the vocalizations to retrieve that toy through the caregiver, which improved vocalizations for that name. In our model, grounding the vocal interaction between the learner and the caregiver in a play scenario thus accelerated the learning of toys' names production.

From those results we suggest the following hypothesis: infant's play with objects in a grounded interaction scenario with a caregiver can accelerate the learning of the vocal production of these objects' name as a result of a goal-directed exploration of the objects. This hypothesis is consistent with experimental data from infant development research. Clerkin et al. (2017) showed that the objects that are frequent in the visual

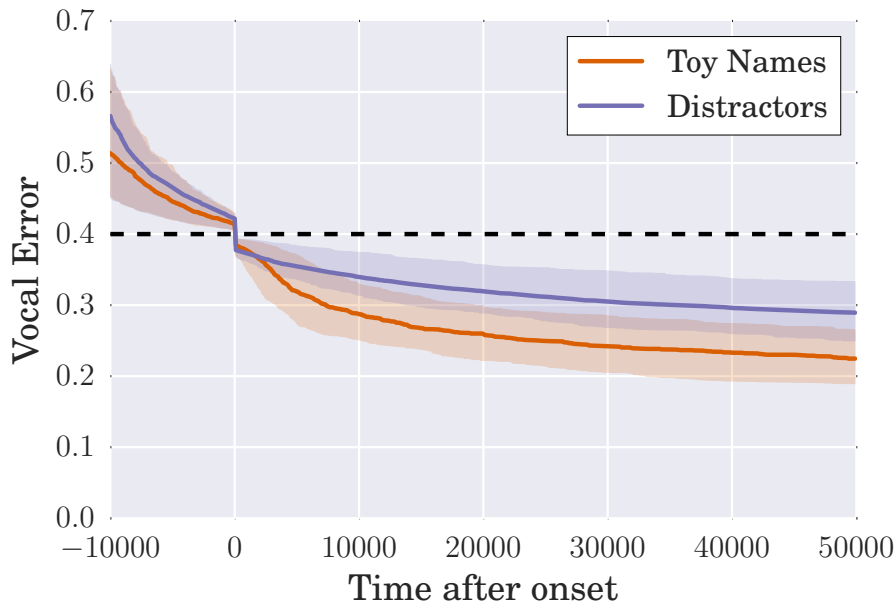


Figure 5.7: Vocal Learning

field of $8\frac{1}{2}$ to $10\frac{1}{2}$ month-old infants are also the objects for which infants acquire the name early. They explain that the particular distribution of object frequency in visual field can help linking the heard label to the good object in a scenario where the caregiver says the name of an object. This data is also consistent with our hypothesis: the most frequent objects in the visual field are the ones that the infant will most often choose goals for, and will trigger caregiver's help when needed by trying to vocalize those toys' names. Infants could thus receive more vocal feedback for those words and learn to produce them earlier. This view also fits with recent data about the body-object interaction measure. In Thill and Twomey (2016), the authors used a measure of the extent to which adults could easily interact with a named item and show that it better predicts the age of acquisition of the name of an item than its concreteness or imageability. In other words, the easier the interaction with an object is to the adult, the sooner its name will be acquired by the infant. The caregiver provides vocal input to the learner, but also provides nonvocal feedback that can help with vocal learning. Indeed, Goldstein et al. (2003) provided evidence that a nonvocal feedback mechanism such as reacting to infant's vocalizations by smiling or touching the infant can shape vocal babbling in real time. Also, a longitudinal study of infant-caregiver interaction Gros-Louis et al. (2014) shows that in a free play scenario, maternal responsiveness to object-directed child vocalizations increases with age from 8 to 14 month-old. In our experiments, the caregiver responds to words that seem close to a toy name by giving the corresponding toy to the agent, and this behavior increases with age as the accuracy of agent's vocalizations increases. Such a

mechanism could also be an important pathways to infant vocal development.

In Forestier and Oudeyer (2016a,c), we studied a tool-use learning agent with a hierarchical sensorimotor architecture, and showed that this architecture improved learning compared to learning one single sensorimotor model combining all sensory spaces. An intrinsic motivation to explore sensorimotor models with high learning progress further increased learning, resulting in developmental trajectories dynamics similar to infant development. In our present study, we implemented a modular architecture of sensorimotor models explored through a goal-babbling procedure. Agents were not given any initial teleological understanding of speech or tool use, nor any hierarchically-organized action sequencing mechanisms. Still, the unified Model Babbling architecture allowed the emergence of behaviors displaying a nested tool-use structure. Those behaviors include reusing movements of the hand to move the stick and movements of the stick to move a toy, or reusing sound trajectories produced with the simulated vocal tract to trigger the help from the caregiver such that she brings the toy within reach. This result suggests that observing infants using tools or asking for help with toys should not necessarily be interpreted as a correlate of capabilities for complex sequencing and planning of actions.

It should be noted that for the agents in our model, involving the caregiver to move toys through vocalizations is a strategy with no special status with respect to the other strategies. This social interaction emerges from the same drive to refine sensorimotor models through goal babbling as in the learning of stick movements to move the toy. The production of sounds that can be understood by the caregiver as toy names to make it react and help can thus be interpreted as an emergent form of social tool use. Words have been considered as social tools in a perspective proposed by (Borghi et al., 2013) combining the embodied-grounded view of cognition where the cognitive processes are seen as constrained by the body, and the extended mind view, allowing the mind to encompass the brain, the body and external devices. The active use of physical tools have been shown to change the representation of space (Iriki et al., 1996; Osiurak et al., 2012). In Borghi et al. (2013), they review experimental evidence showing that the reaching space of a subject can be extended after the use of words, in a setup where an object could be reached with several strategies: with the hand, with a tool, or through the use of a word which triggers an action from another person.

Our model of speech and tool-use development has several limitations. The manual and vocal input from our simulated caregiver is limited compared to a real infant-caregiver interaction scenario. For instance, the vocal input is one of the three toy's name or three distractor words, and each given word is produced without variability, while infant-directed speech shows a high variability in pitch and formants (Fernald et al., 1989; Kuhl et al., 1997). Infants can learn new vocal forms from the diversity of vocalizations in their mother's contingent speech, while mimicking only do not allow it Goldstein and Schwade (2008). We were also limited to the production of vowels as we could not produce consonants with the DIVA vocal synthesizer and our

representation of the vocal sounds with formants do not discriminate consonants. A more accurate model of a vocal tract could be used together with a higher-dimensional representation of vocal sounds such as MFCCs could be used to produce consonants and CV or VCV clusters as has been done in Philippsen (2018). Also, we did not integrate demonstrations of the use and function of tools by a caregiver which is one of the important pathway to learn tool use (Chen et al., 2000).

In our model, the learner imitates the caregiver by randomly choosing any of the previous words produced by the caregiver, including toy names and distractor words. However, the memory and attention of infants are limited, so infants may imitate only the most frequent words or the words produced with infant-directed speech. Also, if the caregiver is not perfectly contingent and reacts to the infant's utterance or action only after a delay or after saying other words to other peers or sometimes do not even react, it might be hard for the infant to link its actions to the answers of the caregiver and understand the contingencies. In our model, those effects could be studied by implementing a limited memory and an attentional mechanism in the agent's cognitive processes. With those mechanisms, the interactive learning scenario could reinforce even more the learning of the toys' names as the learner, by playing with a toy would trigger caregiver's vocal utterances, which could then be imitated more often than other words as the play interaction with the object and the caregiver settles.

In the Model Babbling architecture implemented in this chapter, we used a simple form of intrinsic motivations where the learning agent explores all sensory spaces, irrespective of its learning progress in each space which is sometimes called Random Model Babbling, and similarly all goals in a particular sensory space, which is called Random Goal Babbling. More sophisticated forms of intrinsic motivations could be implemented as in the MACOB algorithm defined in the next chapter (Forestier and Oudeyer, 2016b), by monitoring the progress made to learn each goal and/or each space of goal. This monitoring of learning progress could help focus exploration towards the regions or objects (physical or sounds) on which the agent is making the most progress, and on the long term organize a developmental trajectory tailored to its learning experience.

Finally, we transposed this experiment in a real robotic setup as the topic of internship of Rémy Portelas. Here, the vocal tract is still simulated with the DIVA synthesizer, but the learning agent is embodied in a Poppy Torso robot. It is mounted in front of a playground containing one toy, and the Baxter robot (the caregiver) reacts to the Torso robot's actions and sounds in the same way as in the simulated experiment (see Figure 5.8). The caregiver produces the name of the toy when the learner touches the toy, and replaces the toy within reach of the learner when he produces a sound close enough to the toy's name. Preliminary results shows that the physical Torso robot can learn how to produce the name of the toy through autonomous exploration of the physical scene and vocal sounds together with the imitation of its real caregiver.

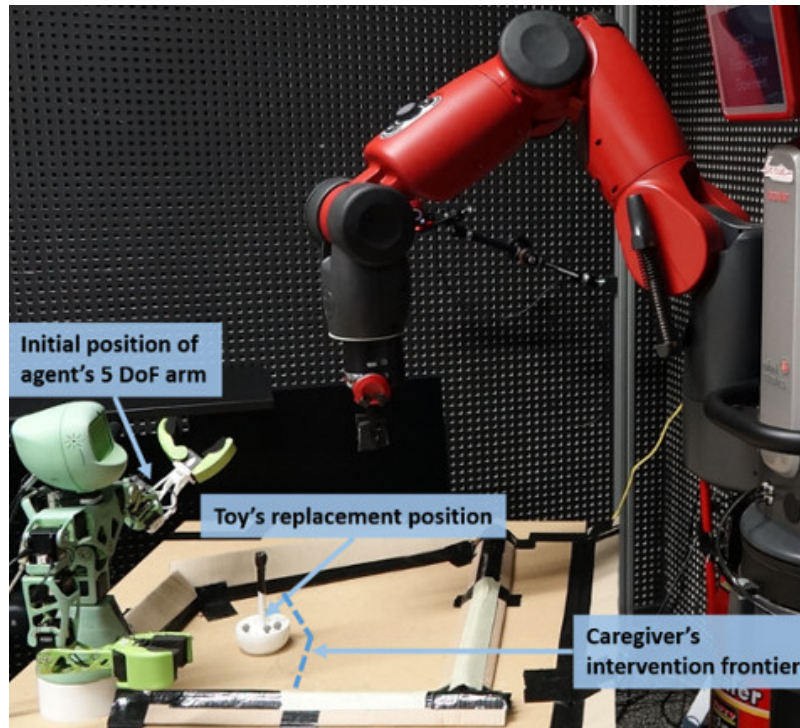


Figure 5.8: The Poppy Torso robot (left) learns to move its arm, control the toy, and vocalize, in interaction with the Baxter robot (right), its caregiver. The simulated experiment was transposed to this real robotic setup during the internship of R my Portelas.

To conclude, our unified robotic model of speech and tool use gives a basis for future research in modeling the manual and vocal interactions between an infant and its caregiver in a grounded naturalistic scenario. From this study, we derived experimental predictions that could drive new experiments with infants and allow us to test and refine the model.

Part II

Intrinsically Motivated Artificial Intelligence

Chapter 6

Modular Active Curiosity-Driven Discovery of Tool Use

Summary

In this chapter, we study algorithms used by a learner to explore high-dimensional structured sensorimotor spaces such as in tool-use environments. In particular, we consider goal babbling architectures that were designed to explore and learn solutions to fields of sensorimotor problems, i.e. to acquire inverse models mapping a space of parameterized sensorimotor effects to a corresponding space of parameterized motor primitives. However, so far these architectures have not been used in high-dimensional spaces of effects. Here, we show the limits of existing goal babbling architectures for efficient exploration in such spaces, and introduce a novel exploration architecture called Model Babbling (MB). MB exploits efficiently a modular representation of the space of parameterized effects, and an active version of Model Babbling (MACOB) further improves learning. These architectures are compared in a simulated experimental setup with an arm that can discover and learn how to move objects using several tools, embedding structured high-dimensional continuous motor and sensory spaces (Forestier and Oudeyer, 2016b).

A major challenge in robotics is to learn sensorimotor models in high-dimensional continuous motor and perceptual spaces. Of particular interest is the acquisition of inverse models which map a space of sensorimotor problems to a space of motor programs that solve them. For example, this could be a robot learning which movements of the arm and hand can push or throw an object in each of several target locations, or which arm movements allow to produce which displacements of several objects potentially interacting with each other, e.g. in the case of tool use. Specifically, acquiring such repertoires of skills through incremental exploration of the environment has been argued to be a key target for life-long developmental learning (Baldassarre and Mirolli, 2013; Cangelosi et al., 2015; Ugur et al., 2015).

To approach this challenge, various works have considered the parameterization of these motor and problem spaces. For example, motor programs can be encoded through Dynamical Movement Primitives parameterized by a vector of real numbers (Stulp et al., 2013). Similarly, it is possible to embed targeted sensorimotor problems (also called space of effects or task space) in a dual parameterized space such as the coordinates of the target object location (Baranes and Oudeyer, 2013; Da Silva et al., 2012; Ude et al., 2010), potentially combined with parameters characterizing the position of obstacles (Stulp et al., 2013).

This dual parameterization is useful for several reasons. First, given a database of experiences associating parameters of motor programs to a set of sensorimotor problems they solve (e.g. the effects they produce), it is possible to use optimization and regression techniques to infer the parameters of motor programs that solve new sensorimotor problems which parameters were not encountered during training (Baranes and Oudeyer, 2013; Da Silva et al., 2012; Kupcsik et al., 2013; Stulp et al., 2013; Ude et al., 2010). Second, it allows efficient data collection leveraging the interactions among sensorimotor problems as achieved in goal babbling exploration (Baranes and Oudeyer, 2013; Fabisch and Metzen, 2014; Rolf et al., 2010) and other related approaches (Kupcsik et al., 2013): when the learner is searching for parameters optimizing one sensorimotor problem (typically using policy search or related stochastic optimization methods (Stulp and Sigaud, 2013)), it will often discover parameters that are improving other sensorimotor problems - and update their current best solutions accordingly (Baranes and Oudeyer, 2013).

Next to approaches that have considered finite sets of parameterized problems (Stulp et al., 2014, 2013), other approaches (Baranes and Oudeyer, 2013; Fabisch and Metzen, 2014; Kupcsik et al., 2013; Moulin-Frier et al., 2013; Rolf et al., 2010) have considered the challenge of autonomous exploration and learning of continuous fields of parameterized problems (e.g. discovering and learning all the feasible displacements of objects and their motor solutions). Among them, the technique of goal babbling (Baranes and Oudeyer, 2013; Fabisch and Metzen, 2014; Rolf et al., 2010), which can be made active (Baranes and Oudeyer, 2013; Fabisch and Metzen, 2014), was shown to be highly efficient for complex tasks such as learning to throw an object in all direction with a flexible fishing rod (Nguyen and Oudeyer, 2014), learning

omnidirectional legged locomotion on slipping surfaces (Baranes and Oudeyer, 2013) or learning to control a robotic pneumatic elephant trunk (Rolf et al., 2010).

However, to our knowledge, results of goal babbling approaches as well as results of other approaches to learning inverse models were so far achieved in relatively low-dimensional spaces of parameterized problems. Furthermore, they were also experimented in sensorimotor spaces with little structure, and in particular have not yet been applied to sensorimotor problems involving tool use.

In this chapter, the primary question we address is: Can goal babbling approaches efficiently drive exploration in high-dimensional structured sensorimotor spaces, such as in tool-use discovery? As we will show, applying them as they exist does not allow an efficient exploration of the sensorimotor space. Rather, we will present a novel algorithmic architecture for exploration, called Model Babbling, that drives sensorimotor data collection by considering a modular representation of the sensorimotor space: instead of considering a flat architecture mapping a motor space to a single high-dimensional space of effects, it considers a set of submodels mapping the motor space to various subspaces of the space of effects. When selected, each of these submodels is explored using the goal babbling approach, and the architecture leverages the fact that exploring one submodel produces data that can improve other submodels.

A secondary issue we study is whether active learning methods can improve the efficiency of this Model Babbling approach. In particular, we present an active Model Babbling architecture, called Modular Active Curiosity-driven mOdel Babbling (**MACOB**), where a measure of empirical learning progress is used by a multi-armed bandit algorithm to select which model to explore (Baldassarre and Mirolli, 2013; Baranes and Oudeyer, 2013; Fabisch and Metzen, 2014). This curiosity-driven exploration algorithm can be related to work using intrinsic motivations in the Reinforcement Learning literature (Baldassarre and Mirolli, 2013; Chentanez et al., 2005; Schmidhuber, 2010).

The study we present is instantiated in a simulated experimental setup¹ with an arm that can discover and learn how to move objects using two tools with different properties. Compared to other work that have studied autonomous tool-use learning (Antunes et al., 2015; Guerin et al., 2013; Stoytchev, 2005; Tikhanoﬀ et al., 2013), this study is original in that it combines 1) considering the problem of how to design efficient exploration algorithms rather than how to design efficient exploitation algorithms that can build compact models from the data collected through exploration; 2) considering a problem of tool-use discovery where tools are objects with initially no special status with respect to other objects, i.e. the robot does not know they are “tools”.

¹Open-source code, notebooks and videos are available on GitHub at <https://github.com/sebastien-forestier/IR0S2016>

6.1 Exploration Architectures

The problem settings for the learning agent is to explore its sensorimotor space and collect data so as to generate a diversity of effects and that this collected database of learning exemplars can be used to build inverse models to be able to reproduce those effects. An agent is described as two independent components: an exploration algorithm and an exploitation algorithm (see Fig. 6.1). The exploration algorithm decides at each iteration which motor command m to explore, and gathers a sensory feedback s to update a database of sensorimotor experiences. We suppose that the motor space M and the sensory space S are continuous and high-dimensional, and that a factorization of S as a product of sensory subspaces that represents the items of the environment can be given to the agent. As detailed below, the exploration algorithm can make use of the current database of sensorimotor experiences to define a coarse but fast surrogate inverse model to orient the exploration process. On the other hand, the exploitation algorithm uses the database built during exploration to generate a potentially more precise inverse model, i.e. to find motor commands to reach sensory goals given by the experimenter based on the explored data. The inverse model of the exploitation algorithm can be built during exploration as an incremental and asynchronous process, or built at the end of exploration as a batch process. Next we describe the exploration architectures, the experimental setup and exploitation architectures.

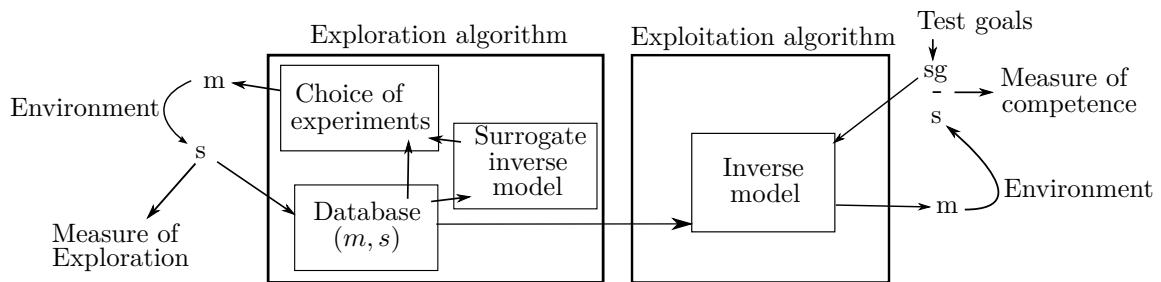


Figure 6.1: Agent's two components: the exploration and exploitation algorithms.

6.1.1 Random Motor Babbling

We first define a control architecture where the agent always chooses random motor commands to try in the environment (RmB, see Algo. 1).

Algorithm 1 Random Motor Babbling (RmB)

Require: Motor space M , Sensory space S

```

1: database  $\leftarrow$  VoidDatabase( $dim(M), dim(S)$ )
2: loop
3:    $m \leftarrow$  RandomMotor( $M$ )
4:    $s \leftarrow$  Environment( $m$ )
5:   Add(database, ( $m, s$ ))
6: return database

```

6.1.2 Random Goal Babbling

In the following exploration architectures the agent performs Goal Babbling. With this method, it self-generates goals in the sensory space and tries to reach them but adds some exploration noise to its motor commands to discover new effects. To generate those goals, different strategies have been studied (Baranes and Oudeyer, 2013). It was shown that estimating the learning progress in different regions of the sensory space and generating the goals where the progress is high leads to fast learning. However, this cannot be applied in a high-dimensional sensory space as a learning progress signal could not be efficiently estimated.

Consequently, we use random goal babbling: goals are randomly generated in the sensory space. This method was nevertheless proven to be highly efficient in complex sensorimotor spaces (Rolf et al., 2010). To perform goal babbling, the agent uses a sensorimotor model that learns a mapping between M and S and provide the inverse inference of a probable motor command m to reach a given sensory goal s_g (see Algo. 2 and 3). The sensorimotor model stores sensorimotor information of the form $(m + \eta, s)$ with $m + \eta$ being the inferred motor parameters to reach the sensory goal, plus Gaussian exploration noise (of standard deviation $\sigma = 0.01$), and $s \in S$ the associated sensory feedback in a sensorimotor database. Section 6.1.5 explains in more detail the two different algorithms that will be used to implement inverse models. We use the Explauto autonomous exploration library (Moulin-Frier et al., 2014) to implement the sensorimotor models and goal babbling. In our implementation, the agent first begins by exploring random motor commands to bootstrap the sensorimotor model until at least 2 distinct sensory points have been reached, and then it starts goal babbling.

Algorithm 2 Random Goal Babbling Step

Require: Sensorimotor model `sm_model`

- 1: $s_g \leftarrow \text{RandomGoal}(S)$
 - 2: $m \leftarrow \text{Inverse}(\text{sm_model}, s_g)$
 - 3: $\eta \leftarrow \text{Gaussian}(\mu = 0, \sigma = 0.01)$
 - 4: $s \leftarrow \text{Environment}(m + \eta)$
 - 5: **return** $(m + \eta, s)$
-

Algorithm 3 Random Goal Babbling Experiment (Baranes and Oudeyer, 2013)

Require: Motor space M , Sensory space S

- 1: `database` \leftarrow `VoidDatabase`($\text{dim}(M), \text{dim}(S)$)
 - 2: `sm_model` \leftarrow `InitializeSensorimotorModel`(M, S)
 - 3: **loop**
 - 4: $(m, s) \leftarrow \text{RandomGoalBabblingStep}(\text{sm_model})$
 - 5: `Update`(`sm_model`, (m, s))
 - 6: `Add`(`database`, (m, s))
 - 7: **return** `database`
-

6.1.3 Model Babbling

We call *flat* exploration architecture the random goal babbling strategy applied to explore directly a mapping between the motor space M and the sensory space S . However, the high-dimensional sensory space (e.g. 93D, see experimental setup after) can be separated into several subspaces to reflect the perception of the different items of the environment (e.g. $p = 15$ subspaces). We thus define a *modular* architecture that explores p sensorimotor models at the same time (one model for each sensory subspace). Each of those modules functions in the same way as a random goal babbling flat architecture, with M as motor space but a specific sensory subspace. However, at each iteration the modular architecture first has to choose the module that will perform goal babbling - pick a random goal in the corresponding sensory subspace. We call this procedure Model Babbling. In a first condition, the babbling module is randomly chosen, which we call Random Model Babbling (See Algo. 4). Once a model is chosen, the agent generates a random goal in the sensory subspace corresponding to that model, infers motor parameters to reach that goal, and adds exploration noise as in the flat architectures. Finally, when motor parameters m

have been executed and feedback s received from the environment, the sensorimotor mappings of all modules are updated with their respective part of s .

Algorithm 4 Modular - Random Model Babbling

Require: Motor space M

Require: Sensory spaces S_i for $i \in \{1..p\}$

```

1: database  $\leftarrow$  VoidDatabase( $dim(M), dim(S)$ )
2: for  $i \in \{1..p\}$  do
3:   sm_model_i  $\leftarrow$  SMMModel( $M, S_i$ )
4: loop
5:   mod_i  $\leftarrow$  RandomModule( $\{1..p\}$ )
6:   ( $m, s$ )  $\leftarrow$  RandomGoalBabblingStep(sm_model_i)
7:   for  $j \in \{1..p\}$  do
8:     Update(sm_model_j, ( $m, Projection(s, S_j)$ ))
9:   Add(database, ( $m, s$ ))
10: return database

```

6.1.4 Active Model Babbling (MACOB)

In strategic learning, different parameterized problems and strategies to solve them are available and the agent learns which strategies are useful for which problems. It was shown in Nguyen and Oudeyer (2012) that an active choice of the outcomes and strategies based on the learning progress on each of them increases learning efficiency compared to a random choice. Also, in Baranes and Oudeyer (2013), the authors develop the SAGG-RIAC architecture of algorithms where the sensory space is automatically split into regions where the learning progress is monitored, and goals are generated in regions where the progress is high. Here, instead of differentiating the learning progress in different regions of a single space, we differentiate it in different sensory spaces.

To implement an active choice of model to explore (Active Model Babbling, see Algo. 5), we first define a measure of interest based on the learning progress of each of the p modules (see Algo. 6). When a module has been chosen to babble, it draws a random goal s_g and finds motor parameters m to reach this goal. The actually reached outcome s in its sensory subspace might be very different from s_g . To measure the progress made to reach s_g , we compare the reached point s with the point s' that was reached for the most similar previous goal s'_g . We define a distance D_{S_i} between two points s and s' in a sensory subspace S_i as the L_2 distance divided by the maximal distance in this sensory subspace, in order to scale this measure across

Algorithm 5 Modular - Active Model Babbling (MACOB)**Require:** Motor space M **Require:** Sensory spaces S_i for $i \in \{1..p\}$

```

1: database  $\leftarrow$  VoidDatabase( $dim(M), dim(S)$ )
2: for  $i \in \{1..p\}$  do
3:   sm_model_i  $\leftarrow$  SMMModel( $M, S_i$ )
4:   i_model_i  $\leftarrow$  InterestModel( $S_i$ )
5:    $I_{mod_i} \leftarrow 0$ 
6: loop
7:    $i \leftarrow$  ChooseModule( $I_{mod_i}$  for  $i \in \{1..p\}$ )
8:    $(m, s) \leftarrow$  RandomGoalBabblingStep(sm_model_i)
9:    $I_{mod_i} \leftarrow$  UpdateInterestModel(i_model_i,  $s_g$ ,
10:                                     Projection( $s, S_i$ ))
11:   for  $j \in \{1..p\}$  do
12:     Update(sm_model_j, ( $m, Projection(s, S_j)$ ))
13:   Add(database, ( $m, s$ ))
14: return database

```

subspaces:

$$D_{S_i}(s, s') = \frac{\|s - s'\|}{\max_{s_1, s_2} \|s_1 - s_2\|} \quad (6.1)$$

We define the interest $I(s_g)$ associated to the goal $s_g \in S_i$:

$$I(s_g) = |D_{S_i}(s_g, s') - D_{S_i}(s_g, s)| \quad (6.2)$$

where s_g and s are the current goal and reached sensory points, and s'_g and s' are the previous goal of that module that is the closest to s_g , and its associated reached sensory point. The interest of a module is initialized at 0 and updated to follow the progress of its goals (with rate $n = 1000$):

$$I_{mod}(t) = \frac{n-1}{n} I_{mod}(t-1) + \frac{1}{n} I(s_g) \quad (6.3)$$

where t is the current iteration: $t \in [1..100000]$.

Finally, we implement a multi-armed bandit algorithm to choose the babbling module at each iteration (Baranes and Oudeyer, 2013; Fabisch and Metzen, 2014). The choice of module is probabilistic and proportional to their interest, with $\epsilon = 10\%$ of random choice to set up an exploration/exploitation tradeoff.

Algorithm 6 Update Interest Model

Require: Interest model `i_model`**Require:** Sensory goal s_g , outcome s

- 1: $(s'_g, s') \leftarrow \text{NearestNeighbor}(\text{goal_database}, s_g)$
 - 2: $I(s_g) = |D_{S_i}(s_g, s') - D_{S_i}(s_g, s)|$
 - 3: $I_{mod_i} \leftarrow \frac{n-1}{n} I_{mod_i} + \frac{1}{n} I(s_g)$
 - 4: $\text{Add}(\text{goal_database}, (s_g, s))$
 - 5: **return** I_{mod_i}
-

6.1.5 Sensorimotor models

Here we describe two algorithms to provide fast, incremental and online forward and inverse model based on a sensorimotor database of motor commands and associated sensory feedback. The first algorithm is the Nearest Neighbor (NN) algorithm, which finds the nearest neighbor of a given point in a database based on a kd-tree search. The forward model is implemented by the following: given a motor command m , the NN algorithm finds the nearest motor command m' in the motor part of the database, and returns the sensory point associated to m' . Also, the inverse of a sensory goal s_g is computed as the motor part m' of the nearest neighbor s' of s_g in the sensory part of the sensorimotor database (see Algo. 7).

Algorithm 7 NN Sensorimotor Model

- 1: **function** INITIALIZE(M, S)
 - 2: `sm_database` \leftarrow VoidDatabase($\text{dim}(M), \text{dim}(S)$)
 - 3: **function** UPDATE((m, s))
 - 4: Add(`sm_database`, (m, s))
 - 5: **function** FORWARD(m)
 - 6: $(m', s') \leftarrow \text{NearestNeighbor}(\text{sm_database}, m)$
 - 7: **return** s'
 - 8: **function** INVERSE(s_g)
 - 9: $(m', s') \leftarrow \text{NearestNeighbor}(\text{sm_database}, s_g)$
 - 10: **return** m'
-

The second algorithm allows to interpolate and extrapolate the forward model around explored points with the Locally Weighted Linear Regression (LWLR, Cleveland and Devlin (1988)). Given a motor command m , LWLR computes a linear regression of the forward model based on the $k = 10$ nearest neighbors of m in the motor part of the database, weighted locally. The weights of the k nearest neighbors of m depends on the distance to m with a Gaussian decreasing function of standard

deviation $\sigma = 0.1$, and LWLR then computes the prediction s_p of m with this local regression (see Algo. 8). On the other hand, the inverse m^* of a sensory goal s_g is found by the minimization of the predicted distance between the reached and goal sensory points as the error function $e(m) = ||Forward(m) - s_g||^2$ with an optimization algorithm (we use the L-BFGS-B algorithm, Byrd et al. (1995)). We limit the number of forward model evaluations (which uses LWLR) to 200.

Algorithm 8 LWLR-BFGS Sensorimotor Model

```

1: function INITIALIZE( $M, S$ )
2:   sm_database  $\leftarrow$  VoidDatabase( $dim(M), dim(S)$ )
3: function UPDATE( $(m, s)$ )
4:   Add(sm_database,  $(m, s)$ )
5: function FORWARD( $m$ )
6:   knns  $\leftarrow$  KNearestNeighbors(sm_database,  $m$ )
7:   weights  $\leftarrow$  GaussianWeights(Distance(knns,  $m$ ))
8:   R  $\leftarrow$  LWLRegression(knns, weights)
9:    $s_p \leftarrow R(m)$ 
10:  return  $s_p$ 
11: function INVERSE( $s_g$ )
12:  error( $m$ ) =  $||FORWARD(m) - s_g||^2$ 
13:   $m^* \leftarrow$  L-BFGS-B-Minimize(error)
14:  return  $m^*$ 

```

6.1.6 Summary of Exploration Architectures

- RmB: Random motor Babbling control (Algo. 1),
- F-NN-RGB: Flat, Nearest Neighbor forward and inverse models, Random Goal Babbling (Algo. 2, 3, 7),
- F-LWLR-RGB: Flat, Locally Weighted Linear Regression forward model and optimization-based inverse model, Random Goal Babbling (Algo. 2, 3, 8),
- M-NN-RMB: Modular, Nearest Neighbor forward and inverse models, Random Model Babbling (Algo. 2, 3, 4, 7),
- M-NN-AMB: Modular, Nearest Neighbor forward and inverse models, Learning Progress based Active Model Babbling (Algo. 2, 3, 5, 6, 7),
- M-LWLR-RMB: Modular, Locally Weighted Linear Regression forward model, optimization-based inverse model, Random Model Babbling (Algo. 2, 3, 4, 8),

- M-LWLR-AMB: Modular, Locally Weighted Linear Regression forward model, optimization-based inverse model, Active Model Babbling (Algo. 2, 3, 5, 6, 8).

6.2 Tool-Use Simulated Environment

We designed a robotic setup where a 2D simulated arm can grasp two sticks that can be used to move some of the out-of-reach objects (see Fig.6.2). The different items in the scene and their interactions are described in the next sections. See Fig.6.2 for a possible state of the environment.

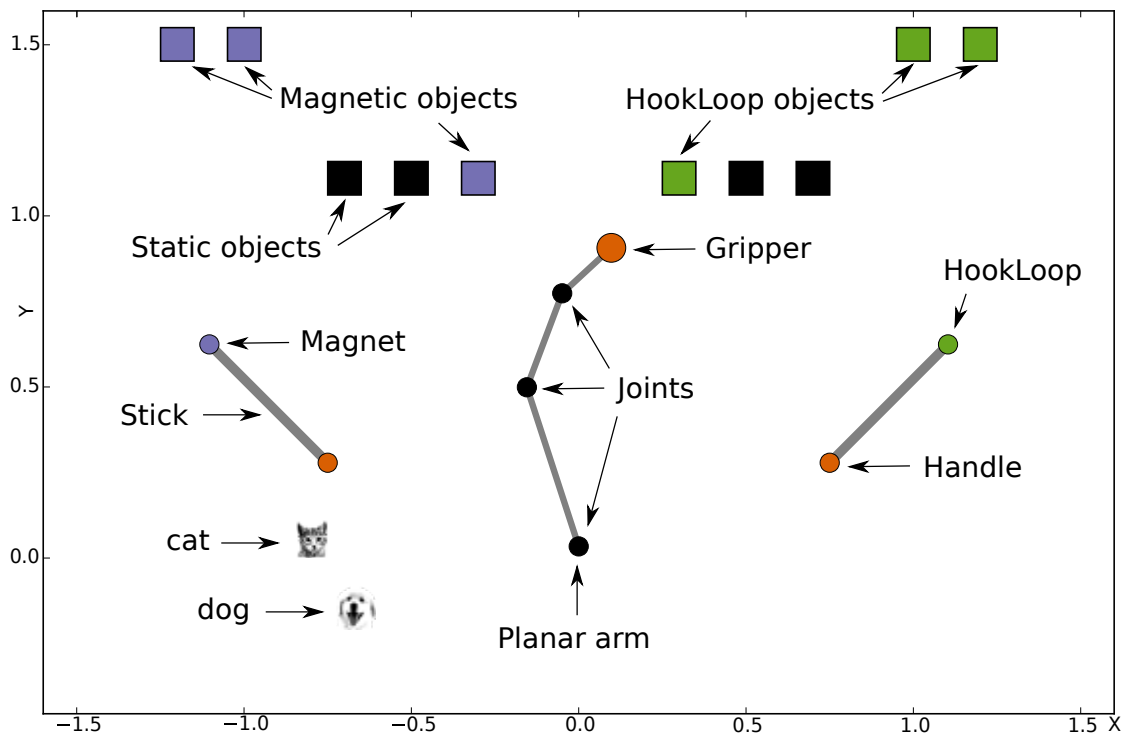


Figure 6.2: A possible state of the environment.

6.2.1 Robotic Arm

The 2D robotic arm has 3 joints plus a gripper located at the end of the arm. Each joint can rotate from $-\pi$ rad to π rad around its resting position, mapped to a standard interval of $[-1, 1]$. The length of the 3 segments of the arm are 0.5, 0.3 and 0.2 so the length of the arm is 1 unit. The resting position of the arm is vertical with joints at 0 rad and its base is fixed at position $(0, 0)$. The gripper g has 2 possible positions: *open* ($g \geq 0$) and *closed* ($g < 0$) and its resting position is *open* (with

$g = 0$). The robotic arm has 4 degrees of freedom represented by a vector in $[-1, 1]^4$. A trajectory of the arm is represented as a sequence vectors.

6.2.2 Motor Control

We use Dynamical Movement Primitives (Ijspeert et al., 2013) to control the arm's movement as this framework allows the production of a diversity of arm's trajectories with few parameters. Each of the 4 arm's degrees-of-freedom (DOF) is controlled by a DMP starting at the resting position of the joint. Each DMP is parameterized by one weight on each of 2 basis functions and one weight specifying the end position of the movement. The weights are bounded in the interval $[-1, 1]$ and allow each joint to fairly cover the interval $[-1, 1]$ during the movement. Each DMP outputs a series of 50 positions that represents a sampling of the trajectory of one joint during the movement. The arm's movement is thus parameterized with 12 weights, represented by the motor space $M = [-1, 1]^{12}$.

6.2.3 Objects and Tools

Two sticks can be grasped by the handle side (orange side) in order to catch an out-of-reach object. The sticks have length 0.5 and are located at positions $(-0.75, 0.25)$ and $(0.75, 0.25)$ as in Fig. 6.2. One stick has a magnet on the end and can catch magnetic objects (represented in blue), and the other stick has a hook-and-loop tape to catch another type of objects (objects represented in green). If the gripper is closed near the handle of one stick (closer than 0.25), this stick is considered grasped and follows the gripper's position and the orientation of the arm's last segment until the gripper opens. In some conditions, we add environmental noise as a Gaussian noise of standard deviation 0.1 added to the (normally equal to 0) angle between the stick and the arm's last segment, different at each of the 50 movement's steps. If the other side of one stick reaches (within 0.25) a matching object (magnetic or hook-and-loop), the object will then follow the end of the stick. Three magnetic objects are located at positions $(-0.3, 1.1)$, $(-1.2, 1.5)$ and $(-1., 1.5)$, so that only one is reachable with the magnetic stick. Three hook-and-loop objects are located at positions $(0.3, 1.1)$, $(1., 1.5)$ and $(1.2, 1.5)$, so that only one is reachable with the hook-and-loop stick. Also, two animals walk randomly following a Gaussian noise of standard deviation 0.01 on X and Y dimensions added at each of the 50 steps of a trial. Finally, four static black squares have also no interaction with other objects. The arm, tools and other objects are reset to their initial state at the end of each iteration.

6.2.4 Sensory Feedback

At the end of the movement, the robot gets sensory feedback representing the trajectory of the different items of the environment during the arm's movement. This

feedback is composed by the position of each item at 3 time points: at steps 17, 33, and 50 during the movement of 50 steps. First, the trajectory of the gripper is represented as a sequence of X and Y positions and aperture (1 or -1) of the gripper (S_{Hand} , 9D). Similarly, the trajectories of the end points of the sticks are sequences of X and Y positions (S_{Stick_1} and S_{Stick_2} , 6D each). Also, the trajectory of each object is a sequence of X and Y positions: S_{Object} with $Object \in \{Magnetic_1, Magnetic_2, Magnetic_3, HookLoop_1, HookLoop_2, HookLoop_3, Cat, Dog, Static_1, Static_2, Static_3, Static_4\}$. Those spaces are all in 6 dimensions ($[-1.5, 1.5]^6$). The total sensory space S has 93 dimensions and corresponds to 15 items.

6.3 Exploitation Architectures

An exploitation architecture generates an inverse model of the environment based on a database of previously explored motor commands and their associated sensory feedback. In this paper, we are both interested in the quality of the exploration databases and in comparing the inverse models built by different combinations of exploration database and exploitation architectures. We evaluate the accuracy of the resulting inverse models to reach points in two spaces of interest, $S_{Magnetic_1}$ and $S_{HookLoop_1}$. Indeed, those spaces represent the only objects that can be moved by one of the sticks as they are not static and not out-of-reach. One set of goals is randomly drawn in the 2D subspace corresponding to the final position of each of the two interesting objects (1000 goals in each).

We define two exploitation architectures generating inverse models: one based on the Nearest Neighbor algorithm (NN, Algo. 7), and one based on the Locally Weighted Linear Regression forward model and an optimization-based inverse model (LWLR, Algo. 8). Given a goal s_g (e.g. $s_g = (0.5, 0.5)$, the final position of the reachable magnetic object), the NN algorithm looks into the explored database, finds the nearest sensory reached point s along the dimensions of the target effect space, and returns its associated motor command m . On the other hand, the LWLR algorithm builds a forward model based on a locally weighted linear regression, and an optimization algorithm (L-BFGS-B) finds the motor command m that minimizes the distance between the prediction of the sensory feedback and the sensory goal.

6.4 Results

We run 100 trials of 100000 iterations with environmental noise and 100 trials without noise, for each of the 7 exploration architectures (thus 14 conditions). We first measure the total exploration of 6D spaces of interest $S_{Magnetic_1}$ and $S_{HookLoop_1}$ after 100000 iterations, and provide results depending on the exploration architecture and environmental noise on the orientations of the sticks. Then, for each of the 1400

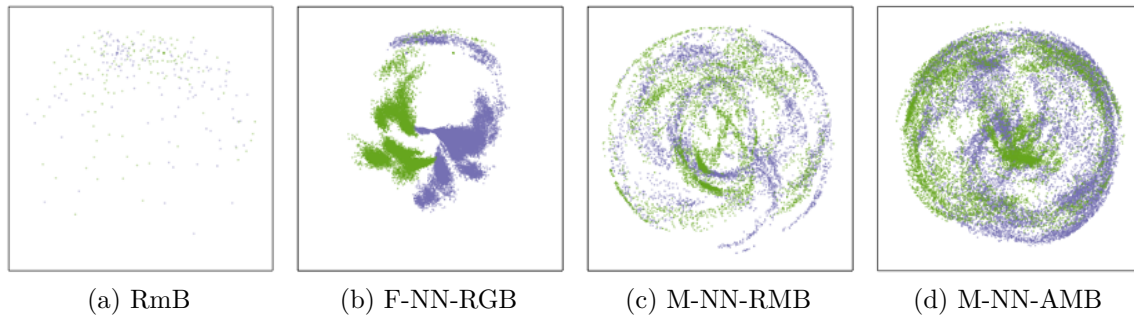


Figure 6.3: Position of the two reachable and movable objects at the end of each of the 100000 iterations, for one trial of some exploration architecture. Blue points: position of reachable magnetic object. Green points: reachable hook-and-loop object.

exploration databases, we test the inverse models generated by the two exploitation architectures in the 2D subspaces of the final position of the two objects of interests, with the same 1000 random goals for each space. We chose those 2D spaces as they represent an interesting effect space from the point of view of the experimenter (as in Fig. 6.3), but the actually learned skills are higher-dimensional (9D for the hand, 6D for each tool and object). We provide a measure of competence of each combination of exploration and exploitation architectures as the median reaching error (the median distance between the goals and actually reached sensory points), both when environmental noise was present and when the environment was deterministic.

6.4.1 Exploration

Examples of Object Exploration

Figure 6.3 shows qualitatively the exploration of the two reachable and movable objects (corresponding to sensory spaces $S_{Magnetic_1}$ and $S_{HookLoop_1}$) for one trial of some exploration architectures, without environmental noise. The blue points are the 2D end positions of the reachable magnetic object, and green points are the end positions of the reachable hook-and-loop object, for the 100000 iterations of an exploration trial. First, the random motor babbling architecture managed to grab the sticks to move one of the object only for a small proportion of the 100000 iterations. Also, only the modular architectures could explore a large proportion of the 2D spaces.

Evolution of Interests in Active Model Babbling

Figure 6.4 shows one example of the evolution of the interest of some of the 15 modules of exploration architecture M-NN-AMB. The first module to make progress

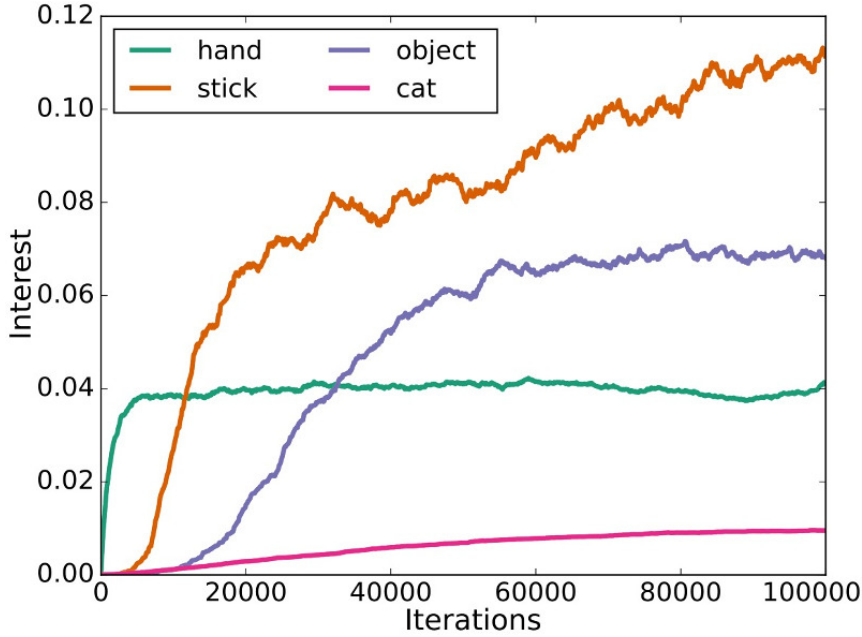


Figure 6.4: Interest of modules along the 100000 iterations, with exploration architecture M-NN-AMB. We show the interest of modules exploring the spaces of the hand, magnetic stick, reachable magnetic object and the cat.

is the module learning to move the hand, and its exploration finds the magnetic stick and thus allows the module corresponding to this stick to make more progress (after 10000 iteration), which exploration finally allows the discovery that this stick can be used to move one of the magnetic objects and make progress on that task (after 20000 iteration). Notably, modules corresponding to unreachable or static objects have an interest strictly equal to 0.

Exploration Measure

The total exploration is measured in $S_{Magnetic_1}$ and $S_{HookLoop_1}$ as the number of cells reached in a discretized grid of 10^6 cells (10 cells on each of the 6 dimensions). For each exploration architecture, we provide in Table 6.1 the median, extrema and quartiles of the number of reached cells (median on 2 spaces times 100 trials). In the following, we give results of non-parametric statistical Mann-Whitney U tests for pairs of conditions.

First of all, the comparison of any of the flat exploration architectures (using NN or LWLR, with or without environmental noise) with any of the modular exploration architectures shows that flat architectures have explored less than modular architectures ($p < 0.05$). The effect is small for example if we compare condition F-LWLR-RGB with environmental noise (median 387 reached cells) with condition M-NN-RMB without noise (median 415). However, the difference is large between

Table 6.1: Exploration of spaces of interest

Exploration architectures	Env. Noise	Min	Q1	Median	Q3	Max
RmB	No	57	67	73	78	93
	Yes	62	75	80	85	100
F-NN-RGB	No	1	1	14	89	380
	Yes	1	1	16	116	746
F-LWLR-RGB	No	98	203	245	294	442
	Yes	182	319	387	486	818
M-NN-RMB	No	285	374	415	456	682
	Yes	356	455	508	563	763
M-NN-AMB	No	88	452	536	668	1380
	Yes	156	431	517	721	1453
M-LWLR-RMB	No	368	512	555	607	801
	Yes	449	574	623	691	906
M-LWLR-AMB	No	456	743	870	1046	1440
	Yes	522	811	987	1153	1752

this flat architecture and the best exploring modular architecture, M-LWLR-AMB with environmental noise (median 987).

Secondly, the comparison of the conditions where only the model babbling choice differs shows that without environmental noise, active model babbling increases exploration with respect to random model babbling. Indeed, architecture M-NN-RMB has explored less (median 415) than architecture M-NN-AMB (median 536, $p < 10^{-23}$), and architecture M-LWLR-RMB also has explored less (median 555) than architecture M-LWLR-AMB (median 870, $p < 10^{-55}$). If we consider environmental noise, the random model babbling architecture using LWLR (median 623) has explored less than the active one (median 987, $p < 10^{-39}$).

6.4.2 Exploitation

The quality of the different inverse models is assessed at the end of the 100000 exploration iterations, by giving random goals in $S_{Magnetic_1}$ and $S_{HookLoop_1}$ and measuring the distance between goals and reached sensory points (without environmental noise). We draw 1000 random sensory goals in each of two spaces of interest, $S_{Magnetic_1}$ and $S_{HookLoop_1}$, and use those same goals for the evaluation of each combination of

Table 6.2: Competence error in spaces of interest

Exploration architecture	Env. Noise	NN	LWLR
RmB	No	0.185	0.711
	Yes	0.307	0.871
F-NN-RGB	No	0.745	1.018
	Yes	1.174	1.253
F-LWLR-RGB	No	0.123	0.171
	Yes	0.376	0.422
M-NN-RMB	No	0.046	0.050
	Yes	0.248	0.261
M-NN-AMB	No	0.035	0.037
	Yes	0.285	0.300
M-LWLR-RMB	No	0.038	0.039
	Yes	0.216	0.227
M-LWLR-AMB	No	0.026	0.026
	Yes	0.215	0.226

exploration and exploitation architectures. Table 6.2 provides the median distance between goals and reached sensory points for each condition (for 2000 points times 100 trials). In the following, we give results of non-parametric statistical Mann-Whitney U tests for pairs of conditions.

Firstly, both if we consider conditions with environmental noise or not, all databases generated by flat exploration architectures and tested by any of the two exploitation architectures show a larger competence error than any of the databases explored with modular architectures and tested with both exploitation architecture ($p < 10^{-100}$). For instance, without environmental noise, the best performing flat condition is F-LWLR-RGB exploited with the NN algorithm, with a median competence error of 0.123, whereas the worst performing modular condition is M-NN-RMB, exploited with the LWLR algorithm, with an error of 0.050.

Secondly, considering only exploration conditions without environmental noise, all databases generated with RMB architectures and tested with any of the two exploitation architectures show a larger competence error than any of the databases generated with AMB and tested with both exploitation architectures ($p < 0.05$). For instance, the median competence error using RMB and the NN algorithm both in exploration and exploitation is 0.046 whereas with AMB it is 0.035. Using LWLR, those errors are 0.039 and 0.026.

6.4.3 Controlling for Random Motor Babbling

In the previous experiments, Flat architectures did only 1 random motor babbling iteration at the beginning of the trial whereas Modular architecture did a small percentage all over the trial duration. Indeed, in our implementation, when targeting a particular sensory space, random motor babbling is used until at least one new sensory effect is achieved in that space, and many modules of the modular representation correspond to a static object, while the Flat representation includes the moving cat and dog. Few initial random experiments could have limited the performances of Flat architectures. If we add 10k iterations of Random Motor Babbling before each experiments to better control for the number of motor babbling experiments, Flat Architectures explore more but still significantly less than Modular Architectures (See Table 6.3). Also, the differences between RMB and AMB are a bit larger with those additional initial random experiments, which might be because it helps to bootstrap learning and to estimate learning progress.

Table 6.3: Exploration, 10k Motor Babbling Bootstrap

Exploration architectures	Env. Noise	Min	Q1	Median	Q3	Max
RmB	No	61	67	72	74	84
F-NN	No	139	164	192	237	439
M-NN-RMB	No	334	368	404	471	569
M-NN-AMB	No	350	585	643	749	1032
F-LWLR	No	188	265	314	370	450
M-LWLR-RMB	No	452	508	527	565	718
M-LWLR-AMB	No	662	786	906	962	1296

6.4.4 Running Longer Experiments

In the previous experiments, using LWLR as the exploitation algorithm did not improve the competence errors. Here we run the same experiments for 300k iterations instead of 100k. Table 6.4 shows the exploration results and Table 6.5 the competence results.

We can see in the competence table that using LWLR in the exploitation algorithm becomes useful after 225k iterations compared to NN. The LWLR local regression indeed gets better with a more dense distribution of points.

Table 6.4: Exploration after 300k iterations

Exploration architectures	Env. Noise	Min	Q1	Median	Q3	Max
RmB	No	138	160	166	175	184
M-NN-RMB	No	799	1015	1107	1221	1384
M-NN-AMB	No	1136	1276	1497	1670	2308
F-LWLR	No	357	431	495	578	761
M-LWLR-RMB	No	1076	1192	1295	1454	1683
M-LWLR-AMB	No	1567	1897	2175	2491	2915

Table 6.5: Competence error in spaces of interest, along 300k iterations

Exploration architecture	Exploit.	75k	150k	225k	300k
RmB	NN	0.2117	0.1542	0.1267	0.1104
	LWLR	0.8413	0.5322	0.3779	0.3132
F-LWLR	NN	0.1313	0.0851	0.0627	0.0492
	LWLR	0.1858	0.1129	0.0786	0.0615
M-NN-RMB	NN	0.0622	0.0319	0.0231	0.0189
	LWLR	0.0705	0.0321	0.0218	0.0176
M-NN-AMB	NN	0.0519	0.0213	0.0149	0.0121
	LWLR	0.0615	0.0217	0.0141	0.0113
M-LWLR-RMB	NN	0.0457	0.0283	0.0220	0.0186
	LWLR	0.0468	0.0274	0.0209	0.0174
M-LWLR-AMB	NN	0.0318	0.0189	0.0147	0.0123
	LWLR	0.0320	0.0180	0.0137	0.0114

6.4.5 Tuning Exploration Noise

In the previous experiments, the exploration noise was set to $\sigma = 0.01$, which is small compared to the parameter values (one percent of those). In order to understand if different algorithms work best for different exploration noise amplitudes, here we also test $\sigma = 0.03$, 0.1 , and 0.3 . Table 6.6 shows the exploration after 100k iteration

depending on exploration noise amplitude. For a noise of 0.1, the exploration of F-NN and F-LWLR are similar. For a noise of 0.03, the exploration of M-NN-AMB and M-LWLR-AMB are similar. Those results show that NN gets as good as LWLR for exploration if we increase the noise amplitude, so in our previous results, the advantage of LWLR exploration might have been due to the inaccuracy of LWLR when few data is available, that acted as an additional exploration noise. Also, the best results for Flat architectures were with a 0.1 noise, and for Modular architectures with $\sigma = 0.03$. With $\sigma = 0.3$, the exploration noise is too large leading to bad exploration results, only slightly better than Random Motor Babbling.

Table 6.6: Exploration, different noise amplitude

Exploration architectures	σ	Min	Q1	Median	Q3	Max
F-NN	0.03	1	178	245	385	769
M-NN-RMB	0.03	484	503	550	596	861
M-NN-AMB	0.03	580	812	950	1144	1349
F-LWLR	0.03	269	323	409	512	702
M-LWLR-RMB	0.03	466	568	613	633	737
M-LWLR-AMB	0.03	609	774	920	975	1379
F-NN	0.1	196	340	418	473	600
M-NN-RMB	0.1	273	328	349	369	412
M-NN-AMB	0.1	268	351	374	405	488
F-LWLR	0.1	260	375	418	455	513
M-LWLR-RMB	0.1	274	335	352	363	393
M-LWLR-AMB	0.1	285	331	350	365	427
F-NN	0.3	155	216	245	263	278
M-NN-RMB	0.3	109	133	140	146	168
M-NN-AMB	0.3	106	116	125	139	156
M-LWLR-RMB	0.3	101	107	113	126	136
M-LWLR-AMB	0.3	100	112	115	121	132

6.4.6 Influence of Modules on Each Other

While targeting a goal in one particular sensory space, such as the space of the hand, new effects can be found in other sensory spaces, such as the one of a tool. Here we study this kind of influence between the exploration in some spaces and the discoveries in others.

When exploring a goal in a particular space, using the Nearest Neighbor sensori-motor model (NN), the closest previously reached point to the goal is used to find a good motor command. We can analyze the origin of that previous point, whether it was discovered when targeting another goal in the same sensory space, in another space, or while doing random motor babbling. Table 6.7 shows for each random goal babbling module mod_j (each column), the number of iterations where mod_j is currently babbling and the nearest neighbor of the goal is a point that was previously reached while exploring mod_i (rows) or by random motor babbling.

Table 6.7: Useful sources of novelty for each module, tolerance 0.25, M-NN-RMB

Useful source	mod_1	mod_2	mod_4
<i>RmB</i>	3799	323	109
mod_1	2502	103	104
mod_2	110	4778	1395
mod_4	39	1316	4994

An interesting case is when mod_2 babbles (tool sensory space), it seems that mod_4 (toy sensory space) helps more than mod_1 (hand sensory space), whereas we would predict that exploring the hand helps more. This may due to the fact that the tolerance to grab the object (in $S_{Magnetic_1}$) was set quite high (0.25), thus exploring diverse goals in $S_{Magnetic_1}$ actually help explore a significant part of S_{stick_1} (with a reduced bottleneck effect).

Table 6.8: Useful sources of novelty for each module, tolerance 0.03, M-NN-RMB

Useful source	mod_1	mod_2	mod_4
<i>RmB</i>	3069	50	88
mod_1	35610	515	42
mod_2	462	18075	283
mod_4	7	157	25

We then reduced that tolerance parameter to 0.03, yielding a smaller tolerance both to grab a tool and to grab a toy with the tool. Table 6.8 shows the influence of modules in that settings. The exploration of S_{stick_1} is now helped by the random goal babbling of mod_1 (515) more than the one of mod_4 (157), which is more consistent with our expectations in this tool-use setup. However, with this hardened setup, finding novelty in $S_{Magnetic_1}$ might take more time.

6.5 Discussion

In this chapter, two new algorithmic architectures were introduced for the incremental exploration of sensorimotor spaces exploiting a modular representation of these spaces. Random Model Babbling selects randomly which model to explore (which is itself explored through goal babbling) and Active Model Babbling uses a multi-armed bandit algorithm to maximize empirical learning progress. In a simulation involving structured continuous high-dimensional motor (12D) and sensory (93D) spaces, we showed that these modular architectures were vastly more efficient than goal babbling methods used with flat representations, for all combinations of inverse models in the exploration and exploitation architectures. In particular, by focusing exploration on relevant parts of the space, modular architectures allowed the learner to discover efficiently how to move various objects using various tools, while flat architectures were not able to discover large parts of the space of effects. We also showed that active model babbling was significantly more efficient than random model babbling, yet the difference was smaller than between modular and flat architectures.

The Model Babbling architecture used in our tool-use setup assumes that a representation of each object is initially given to the learning agent. Given the potential importance of modular representations to address the challenge of incremental learning of skills in high-dimensional spaces, and within a life-long developmental perspective, this work points to the need for algorithmic mechanisms that can generate automatically such representations. In a follow-up work, we study the possibility of learning a representation of the scene directly from pixels (see Appendix B). However, “tool” objects are not given a special status: they are represented in the same way as any other object. Yet, the active model babbling architecture converged to explore preferentially these objects and discovered their actual use as tools.

In chapter 4, the learning agents used a hierarchical organization of modules which was imposed and not built autonomously. A remaining question is how to transform the modular architecture into a hierarchical one where causal dependencies between objects could be represented and leveraged, with the discovery of explicit object categories such as “tools”. A possible approach could be to differentiate “tools” using a relative measure of learning progress, following the approach presented in (Oudeyer et al., 2007, sec. VIII. B. 2) to differentiate the self/body, physical objects and “others”. The statistics of the discoveries made on objects while exploring other objects (see Section 6.4.6) could also help distinguish the relations of causality between objects and uncover the tool-use structure of the environment.

Building a hierarchical representation of related objects in a tool-use environment could benefit exploration. In chapter 4, each inverse model was implemented with a simple nearest neighbor look-up, with an additional Gaussian noise in the lower-level motor space used for exploration of new movements. However, in our tool-use setups, when targeting a new goal for a toy, randomly perturbing a movement known to

grab the tool and reach the toy has many chances to not even grab the tool, such that the tool acts as a bottleneck in exploration. The hierarchy could be used here to implement a structured noise adapted to the sensorimotor hierarchy. Indeed, the Gaussian exploration noise could be applied in the higher-level inverse model, for instance by adding noise to the tool movement and passing down the resulting tool movement to the motor space through the hierarchy of inverse models. The resulting distribution of explored motor commands would be the inverse of the Gaussian distribution through the hierarchy of inverse models, and thus structured by the tool hierarchy, with more chances for new motor commands to grab the tool and yield interesting novel results.

When exploring new goals for a toy that necessitate the use of a tool, modifying only one part of the movement, such as the part after the moment when the tool is grasped, could also improve the efficiency of exploration. In chapter 8, we study this kind of exploration mutations, called Stepping-Stone Preserving Mutations. Another limitation of our experiments here is that we reset all objects to their initial position at each iteration. In a setup where we reset the object to a random position (chapter 4, Forestier and Oudeyer (2016c)), we showed that agents could transfer the knowledge from reaching the object in one position with the tool to reaching the object in a new position, and bootstrap the exploration of that new situation. In chapter 8, the Minecraft Mountain Cart environment involves closed-loop action policies that depend on a high-dimensional real-time context including the current position of the agent and the objects in the map. To handle that case, we implement movement policies with a neural network that takes as input at each time step the current context of the environment and outputs the action to follow. We also study active model babbling algorithms in a real robotic environment, and analyze in more details a simulated tool-use experiment to understand the influence on learning of the representation of goals, the mutation operator, or the distractor objects.

In our Active Model Babbling implementation, the choice of object to explore (goal space) is based on the learning progress to control the objects, and the particular goal for that object is chosen randomly. A more sophisticated choice of goal based on learning progress could be implemented as in single-space goal babbling approaches, e.g. the SAGG-RIAC algorithm of Baranes and Oudeyer (2010a). In the next chapter, we conceptualize intrinsically motivated goal exploration algorithms in a general formal framework (IMGEP), in which Model Babbling is a particular modular architecture.

Chapter 7

Intrinsically Motivated Goal Exploration Processes

Summary

Intrinsically motivated spontaneous exploration is a key enabler of autonomous lifelong learning in human children. It enables the discovery and acquisition of large repertoires of skills through self-generation, self-selection, self-ordering and self-experimentation of learning goals. We present an algorithmic approach called Intrinsically Motivated Goal Exploration Processes (IMGEP) to enable similar properties of autonomous learning in machines. The IMGEP algorithmic architecture relies on several principles: 1) self-generation of goals as fitness functions and selection of goals based on intrinsic rewards; 2) exploration with incremental goal-parameterized policy search and exploitation of the gathered data; 3) systematic reuse of information acquired when targeting a goal for improving towards other goals. We present a particularly efficient form of IMGEP that uses a modular representation of goal spaces as well as intrinsic rewards based on learning progress. IMGEP is a compact and general framework for the exploration of problems with no objective function or where an objective function is hard to define and optimize, while the intrinsically motivated exploration allows an efficient discovery of a diversity of solutions (Forestier et al., 2017).

An extraordinary property of natural intelligence in humans is their capacity for lifelong autonomous learning. Processes of autonomous learning in infants have several properties that are fundamentally different from many current machine learning systems. Among them is the capability to spontaneously explore their environments, driven by an intrinsic motivation to discover and learn new tasks and problems that they imagine and select by themselves (Berlyne, 1966; Gopnik et al., 1999). Crucially, there is no engineer externally imposing one target goal that they should explore, hand providing a curriculum for learning, nor providing a ready-to-use database of training examples. Rather, children self-select their objectives within a large, potentially open-ended, space of goals they can imagine, and they collect training data by physically practicing these goals. In particular, they explore goals in an organized manner, attributing to them values of interestingness that evolve with time, and allowing them to self-define a learning curriculum that is called a developmental trajectory in developmental sciences (Thelen and Smith, 1996). This self-generated learning curriculum prevents infants from spending too much time on goals that are either too easy or too difficult, and allows them to focus on goals of the right level of complexity at the right time. Within this process, the new learned goals/problems are often stepping stones for discovering how to solve other goals of increasing complexity. Thus, while they are not explicitly guided by a final target goal, these mechanisms allow infants to discover highly complex skills. For instance, biped locomotion or tool use would be extremely difficult to learn by focusing only on these goals from the start as the rewards for these goals are typically rare or deceptive.

An essential component of such organized spontaneous exploration is the intrinsic motivation system, also called curiosity-driven exploration system (Gottlieb et al., 2013). In the last decade, a series of computational and robotic models of intrinsically motivated exploration and learning in infants have been developed (Baldassarre and Mirolli, 2013; Oudeyer and Kaplan, 2007), opening new theoretical perspectives in neuroscience and psychology (Gottlieb et al., 2013). Two key ideas have allowed to simulate and predict important properties of infant spontaneous exploration, ranging from vocal development (Forestier and Oudeyer, 2017; Moulin-Frier et al., 2013), to object affordance and tool learning (Forestier and Oudeyer, 2016a,c). The first key idea is that infants might select experiments that maximize an intrinsic reward based on empirical learning progress (Oudeyer et al., 2007). This mechanism would generate automatically developmental trajectories (e.g. learning curricula) where progressively more complex tasks are practiced, learned and used as stepping stones for more complex skills. The second key idea is that beyond selecting actions or states based on the predictive learning progress they provide, a more powerful way to organize intrinsically motivated exploration is to select goals, i.e. self-generated fitness functions, based on a measure of control progress (Baranes and Oudeyer, 2013). Here, the intrinsic reward is the empirical improvement towards solving these goals (Forestier and Oudeyer, 2016a; Oudeyer and Kaplan, 2007), happening through lower-level policy search mechanisms that generate physical actions. The efficiency

of such goal exploration processes leverages the fact that the data collected when targeting a goal can be informative to find better solutions to other goals (for example, a learner trying to achieve the goal of pushing an object on the right but actually pushing it on the left fails to progress on this goal, but learns as a side effect how to push it on the left).

Beyond neuroscience and psychology, we believe these models open new perspectives in artificial intelligence. In particular, algorithmic architectures for intrinsically motivated goal exploration were shown to allow the efficient acquisition of repertoires of high-dimensional motor skills with automated curriculum learning in several robotics experiments (Baranes and Oudeyer, 2013; Forestier and Oudeyer, 2016a). This includes for example learning omnidirectional locomotion or learning multiple ways to manipulate a complex flexible object (Baranes and Oudeyer, 2013). In the previous chapter, we presented and evaluated a modular version of intrinsically motivated goal exploration called Model Babbling.

A first contribution of this chapter is to present a formalization of Intrinsically Motivated Goal Exploration Processes (IMGEP), that is both more compact and more general than these previous models. A second contribution is the design of a modular population-based implementation of these processes, with an object-based modular goal representation and a goal-parameterized policy constructed from a parameterized set of low-level action policies. In this thesis, we study implementations of IMGEP with a population-based policy (POP-IMGEP), but the use of goal-conditioned monolithic policy (GCP-IMGEP) is another possibility, which has recently been extended to a modular goal space (Colas et al., 2018a). We also discuss the relations between the IMGEP architecture and other related exploration frameworks.

7.1 Intrinsically Motivated Goal Exploration Processes

We define a framework for the intrinsically motivated exploration of multiple goals, where the data collected when exploring a goal gives some information to help reach other goals. This framework considers that when the agent performed an experiment, it can compute the fitness of that experiment for achieving any goal, not only the one it was trying to reach. Importantly, it does not assume that all goals are achievable, nor that they are of a particular form, enabling to express complex objectives that do not simply depend on the observation of the end policy state but might depend on several aspects of entire behavioral trajectories (see Box on features, goals and goal spaces). Also, the agent autonomously builds its goals but does not know initially which goals are achievable or not, which are easy and which are difficult, nor if certain goals need to be explored so that other goals become achievable.

7.1.1 Notations and Assumptions

Let's consider an agent that executes continuous **actions** $a \in \mathcal{A}$ in continuous **states** $s \in \mathcal{S}$ of an **environment** E . We consider policies producing time-bounded rollouts through the dynamics $\delta_E(\mathbf{s}_{t+1} \mid \mathbf{s}_{t_0:t}, \mathbf{a}_{t_0:t})$ of the environment, and we denote the corresponding behavioral trajectories $\tau = \{s_{t_0}, a_{t_0}, \dots, s_{t_{end}}, a_{t_{end}}\} \in \mathbb{T}$.

We assume that the agent is able to construct a goal space \mathcal{G} parameterized by g , representing fitness functions f_g giving the fitness $f_g(\tau)$ of an experimentation τ to reach a goal g (see Box on features, goals and goal spaces). Also, we assume that given a trajectory τ , the agent can compute $f_g(\tau)$ for any $g \in \mathcal{G}$.

Given these spaces \mathcal{S} , \mathcal{A} , \mathcal{G} , the agent explores the environment by sampling goals in \mathcal{G} and searching for good solutions to those goals, and learns a **goal-parameterized policy** $\Pi(\mathbf{a}_{t+1} \mid \mathbf{g}, \mathbf{s}_{t_0:t+1}, \mathbf{a}_{t_0:t})$ to reach any goal from any state.

We can then evaluate the agent's exploration and learning efficiency either by observing its behavior and estimating the diversity of its skills and the reached stepping-stones, or by "opening" agent's internal models and policies to analyze their properties.

7.1.2 Algorithmic Architecture

We present Intrinsically Motivated Goal Exploration Processes (IMGEP) as an algorithmic architecture that can be instantiated into many particular algorithms sharing several general principles (see pseudo-code in Architecture 9):

- The agent autonomously builds and samples goals as fitness functions, possibly using intrinsic rewards,
- Two processes are running in parallel: 1) an exploration loop samples goals and searches for good solutions to those goals with the exploration policy; 2) an exploitation loop uses the data collected during exploration to improve the goal-parameterized policy and the goal space,
- The data acquired when exploring solutions for a particular goal is reused to extract potential solutions to other goals.

7.1.3 Goal Exploration

In the exploration loop, the agent samples a goal g , executes its exploration policy Π_ϵ , and observes the resulting trajectory τ . This new experiment τ can be used to:

- compute the fitness associated to goal g ,
- compute an intrinsic reward evaluating the interest of the choice of g ,

Features, Goals and Goal Spaces

In the general case, the agent has algorithmic tools to construct a goal g as any function f_g computable from a state-action trajectory τ , that returns the fitness of τ for achieving the goal.

Given a trajectory $\tau = \{s_{t_0}, a_{t_0}, \dots, s_{t_{end}}, a_{t_{end}}\}$, the agent computes a set of **features** $\varphi_1(\tau), \dots, \varphi_n(\tau)$. Those features are scalars that encode any static or dynamic property of the environment or the agent itself. Those can be the position of a particular object in the environment at the end of the trajectory τ , or the full trajectory of that object, the position of the robot itself, the energy used by the robot for executing a movement, etc. Combinations of these features (e.g. linear) can be added as new features. The features may be given to the agent, or learned, for instance with a variational auto-encoder (see Laversanne-Finot et al. (2018); Péré et al. (2018)).

A **goal** is constructed by defining a fitness function from those features and with the following tools:

- $f_g(\tau) = \varphi_i(\tau)$: the goal g is to maximize feature i , e.g. maximize agent's speed.
- $f_g(\tau) = -\|\varphi_{i-j}(\tau) - p\|$: the goal g is to reach the vector p with features i to j , using a measure $\|\cdot\|$, e.g. move the ball to particular 3D position.
- $f_g(\tau) = \varphi_i(\tau)$ **if** $\varphi_j(\tau) \leq c$ **else** b : the goal g is to maximize feature i while keeping feature j lower than a constant c , e.g. maximize agent's speed while keeping the energy consumption below 10W.
- $f_g(\tau) = f_{g_1}(\tau)$ **if** $f_{g_2}(\tau) < f_{g_3}(\tau)$ **else** $f_{g_4}(\tau)$: goals can be combined to form more complex constrained optimization problems, e.g. move the ball to follow a target while not getting too close to the walls and holes and minimizing the energy spent.

A **goal space** is a set of goals parameterized by a vector. For instance, the values p , c , and b in the above definition of goals can be used as parameters to form goal spaces. In the experiments of this paper, we define goal spaces with a parameter p representing the goal position of an object in the environment. Each object k defines a goal space \mathcal{G}^k containing goals g_p of the form $f_{g_p}(\tau) = -\|\varphi_{I_k}(\tau) - p\|$ where p is the goal position of the object and I_k are the indices of features representing the X, Y and Z position of the object k at the end of a trajectory.

Architecture 9 Intrinsically Motivated Goal Exploration Process (IMGEP)

Require: Action space \mathcal{A} , State space \mathcal{S}

- 1: Initialize knowledge $\mathcal{E} = \emptyset$
 - 2: Initialize goal space \mathcal{G} and goal policy Γ
 - 3: Initialize policies Π and Π_ϵ
 - 4: Launch asynchronously the two following loops:
 - 5: **loop** ▷ Exploration loop
 - 6: Choose goal g in \mathcal{G} with Γ
 - 7: Execute a roll-out of Π_ϵ , observe trajectory τ
 ▷ From now on $f_{g'}(\tau)$ can be computed to estimate the fitness of the current experiment τ for achieving any goal $g' \in \mathcal{G}$
 - 8: Compute the fitness $f = f_g(\tau)$ associated to goal g
 - 9: Compute intrinsic reward $r_i = IR(\mathcal{E}, g, f)$ associated to g
 - 10: Update exploration policy Π_ϵ with $(\mathcal{E}, g, \tau, f)$ ▷ e.g. fast incremental algo.
 - 11: Update goal policy Γ with $(\mathcal{E}, g, \tau, f, r_i)$
 - 12: Update knowledge \mathcal{E} with (g, τ, f, r_i)
 - 13: **loop** ▷ Exploitation loop
 - 14: Update policy Π with \mathcal{E} ▷ e.g. batch training of deep NN, SVMs, GMMs
 - 15: Update goal space \mathcal{G} with \mathcal{E}
 - 16: **return** Π
-

- update the goal policy (sampling strategy) using this intrinsic reward,
- update the exploration policy Π_ϵ with a fast incremental learning algorithm,
- update the learning database \mathcal{E} .

Then, asynchronously, this learning database \mathcal{E} can be used to learn a target policy Π with a slower or more computationally demanding algorithm, but on the other end resulting in a more accurate policy. The goal space may also be updated based on this data.

7.1.4 Intrinsic Rewards

In goal exploration, a goal $g \in \mathcal{G}$ is chosen at each iteration. \mathcal{G} may be infinite, continuous and of high-dimensionality, making the choice of goal important and non-obvious. Indeed, even if the fitness function $f_{g'}(\tau)$ may give information about the fitness of a trajectory τ to achieve many goals $g' \in \mathcal{G}$, the policy leading to τ has been chosen with the goal g to solve in mind, thus it may not give as much information about other goals than the execution of another policy chosen when targeting other goals.

Intrinsic rewards provide a mean for the agent to self-estimate the expected interest of exploring particular goals for learning how to achieve all goals in \mathcal{G} . An intrinsic reward signal r_i is associated to a chosen goal g , and based on a heuristic (denoted IR) such as outcome novelty, progress in reducing outcome prediction error, or progress in competence to solve problems (Oudeyer and Kaplan, 2007).

In the experiments of this thesis, we use intrinsic rewards based on measuring the competence progress towards the self-generated goals, which has been shown to be particularly efficient for learning repertoires of high-dimensional robotics skills (Baranes and Oudeyer, 2013). Figure 7.1 shows a schematic representation of possible learning curves and the exploration preference of an agent with intrinsic rewards based on learning progress.

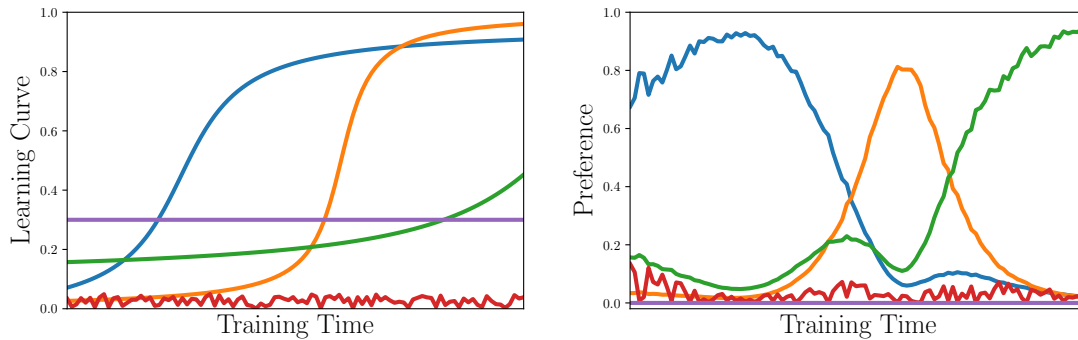


Figure 7.1: Schematic representation of possible learning curves for different goals and the associated exploration preference for an agent with intrinsic rewards based on learning progress. Left: schematic learning curves associated to 5 imaginary goals: the y axis represents the competence of the agent to achieve the goal (1 is perfect, 0 is chance level), and the x axis is training time on a goal. The blue, orange and green curves represent achievable goals, for which agent’s competence increases with training, at different rates, and saturates after a long training time. The purple curve represents a goal on which the agent always has the same competence, with no progress. The red curve is the learning curve on an unreachable goal, e.g. moving an uncontrollable object. Right: exploration preference of an agent with a learning progress heuristic (competence derivative) to explore the 5 goals defined by the learning curves. At the beginning of exploration, the agent makes the most progress on goal blue so it prefers to train on this one, and then its preference will shift towards goals orange and green. The agent is making no progress on goal purple so will not choose to explore it, and goal red has a noisy but low estimated learning progress.

7.2 Modular Population-Based IMGEP

We define here a particular architecture corresponding to the case where the goal space \mathcal{G} is modular and the goal-parameterized policy Π is population-based. Also, we consider that the starting state s_{t_0} of a trajectory is characterized by a parameter vector c called **context** and that the trajectory τ is characterized by a descriptor o_τ called **outcome**, which can be computed by the agent from τ at any time. In the following sections we detail the algorithmic ingredients used in modular population-based IMGEP. Figure 7.2 summarizes the different components of the architecture, and the pseudo-code is provided in Architecture 10.

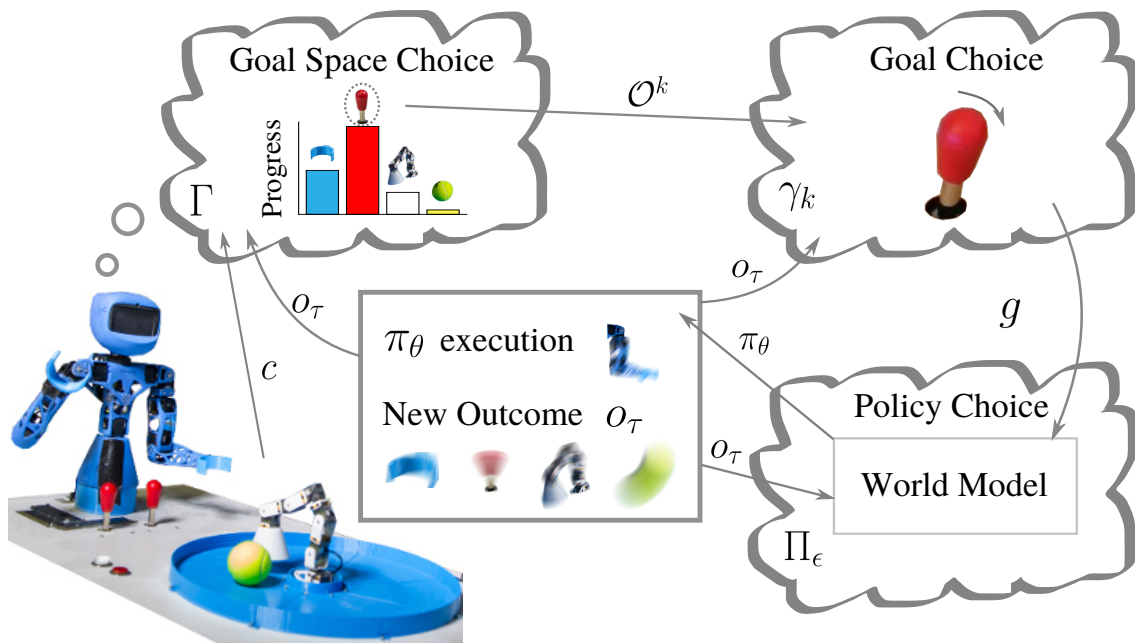


Figure 7.2: Summary of our modular population-based IMGEP implementation. At each iteration, the agent observes the current context c and chooses a goal space to explore based on intrinsic rewards (the learning progress to move each object) with Γ . Then a particular goal g for the chosen object is sampled with γ_k , for instance to push the left joystick to the right. The agent chooses the best policy parameters θ to reach this goal, with the exploration meta-policy Π_ϵ , and potentially using an internal model of the world. The agent executes policy π_θ , observes the trajectory τ and compute the outcome o_τ encoding the movement of each object. Finally, each component is updated with the result of this experiment.

Architecture 10 Modular Population-Based IMGEP**Require:** Action space \mathcal{A} , State space \mathcal{S} , Context space \mathcal{C} , Outcome space \mathcal{O}

- 1: Initialize knowledge $\mathcal{E} = \emptyset$
- 2: Initialize goal space \mathcal{G} , goal policies γ_k and goal space policy Γ
- 3: Initialize meta-policies Π and Π_ϵ
- 4: Launch asynchronously the two following loops:
- 5: **loop** ▷ Exploration loop
- 6: Observe context c
- 7: Choose goal space \mathcal{G}^k with Γ
- 8: Choose goal g in \mathcal{G}^k with γ_k
- 9: Choose policy parameters θ to explore g in context c with Π_ϵ
- 10: Execute a roll-out of π_θ , observe trajectory τ
- 11: Compute outcome o_τ from trajectory τ
 - ▷ From now on, $f_{g'}(\tau)$ can be computed to estimate the fitness of the experiment τ for achieving any goal $g' \in \mathcal{G}$
- 12: Compute the fitness $f = f_g(\tau)$ associated to goal g
- 13: Compute intrinsic reward $r_i = IR(\mathcal{E}, c, g, \theta, o_\tau, f)$ associated to g in context c
- 14: Update exploration meta-policy Π_ϵ with $(\mathcal{E}, c, \theta, \tau, o_\tau)$ ▷ e.g. fast incr. algo.
- 15: Update goal policy γ_k with $(\mathcal{E}, c, g, o_\tau, f, r_i)$
- 16: Update goal space policy Γ with $(\mathcal{E}, c, k, g, o_\tau, f, r_i)$
- 17: Update knowledge \mathcal{E} with $(c, g, \theta, \tau, o_\tau, f, r_i)$
- 18: **loop** ▷ Exploitation loop
- 19: Update meta-policy Π with \mathcal{E} ▷ e.g. batch training of DNN, SVM, GMM
- 20: **return** Π

7.2.1 Goal Construction with Object Modularity

In the IMGEP architecture, the agent builds and samples goals autonomously. Here, we consider the particular case where the agent builds several goal spaces that correspond to moving each object in the environment.

We define the outcome $o_\tau \in \mathcal{O}$ of an experiment τ as the features of the movement of all objects, so that $\mathcal{O} = \prod_k \mathcal{O}^k$ where $o_\tau^k \in \mathcal{O}^k$ are the features of object k . Those features come from a perceptual system that may be given or learned. From feature space \mathcal{O}^k , the agent can autonomously generate a corresponding goal space \mathcal{G}^k that contains goals g as fitness functions of the form $f_g(\tau) = -\|g - o_\tau^k\|_k$. The norm $\|\cdot\|_k$ is a distance in the space \mathcal{O}^k , which can be normalized to be able to compare the fitness of goals across goal spaces. The goal space is thus modular, composed of several object-related subspaces: $\mathcal{G} = \bigcup_k \mathcal{G}^k$.

With this setting, goal sampling is hierarchical in the sense that the agent first chooses a goal space \mathcal{G}^k to explore with a goal space policy Γ and then a particular goal

$g \in \mathcal{G}^k$ with the corresponding goal policy γ_k . Those two levels of choice can make use of self-computed intrinsic rewards r_i (see Sec. 7.1.4). This modular implementation of IMGEP was called Model Babbling in the previous chapter (Forestier and Oudeyer, 2016b).

Given an outcome o_τ , the fitness $f_g(\tau)$ can thus be computed by the agent for all goals $g \in \mathcal{G}$ and at any time after the experiment τ . For instance, if while exploring the goal of moving object A to the left, object B moved to the right, that outcome can be taken into account later when the goal is to move object B.

7.2.2 Population-Based Meta-Policies Π and Π_ϵ

In this version of the IMGEP framework, the goal-parameterized policy Π is population-based: it is built from a set of low-level policies π_θ parameterized by $\theta \in \Theta$, and a meta-policy $\Pi(\boldsymbol{\theta} \mid \mathbf{g}, \mathbf{c})$ which, given a goal and context, chooses the best policy π_θ to achieve the goal g . The policies $\pi_\theta(\mathbf{a}_{t+1} \mid \mathbf{s}_{t_0:t+1}, \mathbf{a}_{t_0:t})$ can be implemented by stochastic black-box generators or small neural networks (see next chapter).

During the goal exploration loop, the main objective consists in collecting data that covers well the space of goals: finding θ parameters that yield good solutions to as many goals as possible. The **exploration meta-policy** $\Pi_\epsilon(\boldsymbol{\theta} \mid \mathbf{g}, \mathbf{c})$ is learned and used to output a distribution of policies π_θ that are interesting to execute to gather information for solving in context c the self-generated goal g and goals similar to g . To achieve the objective of collecting interesting data, the exploration meta-policy Π_ϵ must have fast and incremental updates. As the aim is to maximize the coverage of the space of goals, being very precise when targeting goals is less crucial than the capacity to update the meta-policy quickly and incrementally. In our experiments, the exploration meta-policy $\Pi_\epsilon(\boldsymbol{\theta} \mid \mathbf{g}, \mathbf{c})$ is implemented as a fast memory-based nearest neighbor search with a kd-tree.

On the contrary, the purpose of the **target meta-policy** Π is to be used in exploitation mode: later on, it can be asked to solve as precisely as possible some goals g with maximum fitness. As the training of this meta-policy can be done asynchronously from data collected by the goal exploration loop, this allows the use of slower training algorithms, possibly batch, that might generalize better, e.g. using Gaussian mixture models, support vector regression or (deep) neural networks. These differences justify the fact that IMGEP uses in general two different representations and learning algorithms for Π_ϵ and Π . This two-level learning scheme has similarities with the Complementary Learning Systems Theory used to account for the organization of learning in mammalian brains (Kumaran et al., 2016).

7.3 Relations to Related Work

Early models of intrinsically motivated reinforcement learning (also called curiosity-driven learning) have been used to drive efficient exploration in the context of target tasks with rare or deceptive rewards (Barto, 2013; Schmidhuber, 1991b) or in the context of computational modeling of open-ended unsupervised autonomous learning in humans (Kaplan and Oudeyer, 2004; Oudeyer et al., 2007). Reviews of the historical development of these methods and their links with cognitive sciences and neuroscience can be found in Baldassarre and Mirolli (2013); Gottlieb et al. (2013); Oudeyer et al. (2016).

Several lines of results have shown that intrinsically motivated exploration and learning mechanisms are particularly useful in the context of learning to solve reinforcement learning problems with sparse or deceptive rewards. For example, several state-of-the-art performances of Deep Reinforcement Learning algorithms, such as letting a machine learn how to solve complex video games, have been achieved by complementing the extrinsic rewards (number of points won) with an intrinsic reward pushing the learner to explore for improving its predictions of the world dynamics (Bellemare et al., 2016; Houthoof et al., 2016). An even more radical approach for solving problems with rare or deceptive extrinsic rewards has been to completely ignore extrinsic rewards, and let the machine explore the environment for the sole purpose of learning to predict the consequences of its actions (Oudeyer et al., 2007; Schmidhuber, 1991b), to achieve self-generated goals (Baranes and Oudeyer, 2013; Oudeyer and Kaplan, 2007), or to generate novel outcomes (Lehman and Stanley, 2011a). This was shown for example to allow agents to learn to play some video games without ever observing the extrinsic reward (Pathak et al., 2017).

Some approaches to intrinsically motivated exploration have used intrinsic rewards to value visited actions and states through measuring their novelty or the improvement of predictions that they provide, e.g. Dayan and Sejnowski (1996); Oudeyer et al. (2007); Schmidhuber (1991b); Sutton (1990) or more recently Bellemare et al. (2016); Houthoof et al. (2016); Pathak et al. (2017). However, organizing intrinsically motivated exploration at the higher level of goals, by sampling goals according to measures such as competence progress (Oudeyer and Kaplan, 2007), has been proposed and shown to be more efficient in contexts with high-dimensional continuous action spaces and strong time constraints for interaction with the environment (Baranes and Oudeyer, 2013).

Several proposed methods are related to IMGEP, including Gregor et al. (2016), Dosovitskiy and Koltun (2016) and Kulkarni et al. (2016), however they have considered notions of goals restricted to the reaching of states or direct sensory measurements, did not consider goal-parameterized rewards that can be computed for any goal, used different intrinsic rewards, and did not evaluate these algorithms in robotic setups. The notion of auxiliary tasks is also related to IMGEP in the sense that it allows a

learner to acquire tasks with rare rewards by adding several other objectives which increase the density of information obtained from the environment (Jaderberg et al., 2016; Riedmiller et al., 2018). Another line of related work (Srivastava et al., 2013) proposed a theoretical framework for automatic generation of problem sequences for machine learners, however it has focused on theoretical considerations and experiments on abstract problems.

Several strands of research in robotics have presented algorithms that instantiate such intrinsically motivated goal exploration processes (Baranes and Oudeyer, 2010a; Rolf et al., 2010), using different terminologies such as contextual policy search (Kupcsik et al., 2017; Queißer et al., 2016), or formulated within an evolutionary computation perspective such as Novelty Search (Lehman and Stanley, 2011a) or Quality Diversity (Cully et al., 2015; Cully and Demiris, 2017) (see next sections). In the previous chapter, we implemented a modular population-based version of IMGEP that we called Model Babbling (Forestier and Oudeyer, 2016b). We evaluated several variants of Model Babbling: we called Random Model Babbling a variant where the goal space is chosen randomly and goals are chosen randomly in the goal space and Active Model Babbling one where the goal space is chosen based on the learning progress to control each object. Both implementations are instances of IMGEP as the goal spaces are generated autonomously from the sensory spaces and no “expert knowledge” has been given to the algorithm.

In machine learning, the concept of curriculum learning (Bengio et al., 2009) has most often been used in the context of training neural networks to solve prediction problems. Many approaches have used hand-designed learning curriculum (Sutskever and Zaremba, 2014), but recently it was shown how learning progress could be used to automate intrinsically motivated curriculum learning in LSTMs (Graves et al., 2017). However, these approaches have not considered a curriculum learning of sets of reinforcement learning problems, which is central in the IMGEP framework formulated with goals as fitness functions, and assumed the pre-existence of a database with learning exemplars to sample from. In recent related work, Matiisen et al. (2017) studied how intrinsic rewards based on learning progress could also be used to automatically generate a learning curriculum with discrete sets of reinforcement learning problems, but did not consider high-dimensional modular problem spaces. The concept of “curriculum learning” has also been called “developmental trajectories” in prior work on computational modeling of intrinsically motivated exploration (Oudeyer et al., 2007), and in particular on the topic of intrinsically motivated goal exploration (Baranes and Oudeyer, 2013; Forestier and Oudeyer, 2017).

The concepts of goals and of learning across goals have been introduced in machine learning in Kaelbling (1993) with a finite set of goals. Continuous goals were used in Universal Value Function Approximators (Schaul et al., 2015), where a vector describing the goal is provided as input together with the state to the neural network of the policy and of the value function. However, in these works the goals are not modular, and are considered extrinsic to the agent, with extrinsic rewards that can

contain expert knowledge about the tasks being learned. The learning problem is not formulated as an autonomous learning problem where the agent has to explore the most diverse set of states and skills on its own. Another work integrates intrinsic rewards with an extension of Universal Value Function Approximators (Colas et al., 2018a). This is a particular implementation of the IMGEP architecture, that we call GCP-IMGEP, using a unique monolithic (multi-task multi-goal) policy network, that learns from on a replay buffer filled with rollouts on task and goals of high learning progress. Also, using a population-based intrinsically motivated agent within the IMGEP architecture can help bootstrap a deep RL agent (Colas et al., 2018b). Filling the replay buffer of a deep RL agent with exploratory trajectories collected by an IMGEP algorithm kick-starts the RL agent by enhancing its exploratory abilities. It combines the efficient exploration of population-based IMGEP agents with the efficient fine tuning of policies offered by deep RL agents with a function approximator based on gradient descent.

7.3.1 IMGEP and Novelty Search

In Novelty Search evolutionary algorithms, no objective is given to the optimization process, which is driven by the novelty or diversity of the discovered individuals (Lehman and Stanley, 2011a). In this implementation, an archive of novel individuals is built and used to compute the novelty of the individuals of the current generation of the evolutionary algorithm. If the novelty of a new individual is above a threshold, it is added to the archive. Different measures of novelty can be used, a simple one being the average distance of the individual to its closest neighbors in the archive, the distance being measured in a behavioral space defined by the user. Then, to generate the population of the next generation, the individuals with a high measured novelty are reused, mutated or built upon.

Although designed in an evolutionary framework, the Novelty Search (NS) algorithm can be framed as a population-based IMGEP implementation, assuming that the behavioral space and its distance measure can be self-generated by the algorithm. Indeed, we can define an IMGEP goal space based on the NS behavioral space, with each behavior in that space generating the corresponding goal of reaching that behavior, with a fitness function defined as the negative distance between the target behavior and the reached behavior. In IMGEP, if the goal g (defining the target behavior) is chosen randomly, the algorithm can then reuse the previous reached behaviors that give the highest fitness to reach the current goal g , which are the closest reached points in the behavioral space. The key similarity between our population-based implementations of IMGEP and Novelty Search is that the previous behavior the closest to the current random target behavior is a behavior with high novelty on average. Indeed, a random point in a space is more often closer to a point at the frontier of the explored regions of that space which is thus a high-novelty point. Randomly exploring behaviors or mutating high-novelty behaviors are therefore

efficient for the same reasons.

Abandoning the external objectives and focusing on the novelty of the behaviors in Lehman and Stanley (2011a) can be seen in the lens of the IMGEP framework as embracing all self-generated objectives.

7.3.2 IMGEP and Quality Diversity

The Novelty Search approach stems from the fact that in many complex optimization problems, using a fitness function to define a particular objective and relying only on the optimization of this function do not allow the discovery of the objective as unknown complex successive stepping-stones need to be reached before the final objective can be approached. Relying on novelty allows to reach stepping-stones and build upon them to explore new behaviors even if the objective does not get closer. However, when the behavioral space is high-dimensional, pursuing the final objective is still useful to drive exploration together with the search for novelty (Cuccu and Gomez, 2011). The Quality Diversity approach combines the search for Diversity from Novelty Search approaches and the use of an external objective function to ensure the Quality of the explored individuals (Cully and Demiris, 2017; Lehman and Stanley, 2011b; Mouret and Clune, 2015).

In the MAP-Elites algorithm (Mouret and Clune, 2015), the behavioral space is discretized into a grid of possible behaviors, and a fitness function is provided to assess the quality of individuals according to a global objective. Each new individual is assigned to a behavioral cell in this grid and is given a quality value with the quality function. The population of the next generation of the evolutionary algorithm is mutated, in its simplest version, from a random sample of the set of the best quality individual of all cells. In more sophisticated versions, the parents used for evolving the next generation are selected based on their quality, the novelty of the cells, or a tradeoff between quality and novelty.

In the applications of this algorithm, the fitness function is an extrinsic objective. For instance, in Cully et al. (2015) robot controllers are evolved to find efficient robot walking behaviors. The fitness function given to the algorithm is the speed of the robot, while the descriptors of a behavior can be the orientation, displacement, energy used, deviation from a straight line, joint angles, etc. The algorithm thus tries to find efficient walking behaviors for each behavioral set of constraints.

The concept of Quality Diversity algorithms is thus different from the concept of intrinsically motivated exploration, however Quality Diversity algorithms could be used with a fitness function that is intrinsically generated by the algorithm. In any case, the functioning of the algorithm given this fitness function can also be seen as a population-based implementation of the IMGEP framework. Indeed, each cell of the behavioral grid can generate one different IMGEP goal with a particular fitness function returning the quality of the individual if its behavior falls into that cell and zero otherwise. In MAP-Elites (Mouret and Clune, 2015), the next generation of

individuals is mutated from a random sample of elites (the best quality individual of each non-void cell). In an IMGEP settings with those goals, the MAP-Elites sampling is equivalent to selecting a random goal from the set of goals that had a non-zero fitness in the past. When such a goal is selected, the new IMGEP exploration experiment then reuse, in its simplest version, the sample with the best fitness for that goal, which corresponds to the elite.

In the Novelty Search, Quality Diversity and IMGEP implementations, the key mechanisms that makes exploration efficient are 1) a diversity of solutions continue to be explored even if they seem non-optimal, and 2) when exploring solutions to a given region/cell/goal, the algorithm can find solutions to other regions/cells/goals, which are recorded and can be leveraged later.

7.3.3 IMGEP and Reinforcement Learning

In our setting, the fitness functions f_g have two particularities in comparison with the concept of “reward function” as often used in the RL literature. The first particularity is that these fitness functions are computed based on the trajectory τ resulting from the execution of policy Π , and thus consider the whole interaction of the agent and its environment during the execution of the policy, for instance taking into account the energy used by the agent or the trajectory of an object. Therefore they are not necessarily Markovian if one considers them from the perspective of the level of state transitions s_i .

The second particularity is that since the computation of the fitness $f_g(\tau)$ is internal to the agent, it can be computed any time after the experiment and for any goal $g \in \mathcal{G}$, not only the particular goal that the agent was trying to achieve. Consequently, if the agent stores the observation τ resulting from the exploration of a goal p_1 , then when later on it self-generates goals g_2, g_3, \dots, g_i it can compute, without further actions in the environment, the associated fitness $f_{g_i}(\tau)$ and use this information to improve over these goals g_i . This property is essential as it enables direct reuse of data collected when trying to achieve a goal for later exploring other goals. It is leveraged for curriculum learning in Intrinsically Motivated Goal Exploration Processes.

7.4 Discussion

In this chapter, we defined a formal framework for an exploration architecture called Intrinsically Motivated Goal Exploration Processes (IMGEP). This framework enables a unified description of various related algorithms that share several principles: exploration is driven by self-generated goals, exploring towards a goal gives information that can be reused to improve solutions for other goals, and intrinsic rewards can help the construction and selection of goals. We provided a particular implementation

of IMGEP that formalizes the Model Babbling algorithm described in the previous chapter (Forestier and Oudeyer, 2016b), with spatial modularity: the agent generates one goal space for each object in the environment, and population-based policies: the agent explores a parameterized set of low-level policies.

The IMGEP framework is both compact and general. The goals are defined through fitness functions and therefore can represent any kind of objective that can be computed from the information stored and available to the agent. The policies can be implemented by any algorithm that can learn a function that takes a goal as input and outputs actions to explore this goal, such as a monolithic neural network (Colas et al., 2018a) or a population-based policy (Forestier and Oudeyer, 2016b).

The architecture can also represent algorithms of the Novelty Search evolutionary framework, which do not rely on optimizing an external objective but rather evolve a set of solutions as diverse as possible. The Quality Diversity optimization framework can also be framed as an IMGEP if we assume that the fitness function giving the quality of individuals can be generated autonomously by the learner. Otherwise, Quality Diversity implementations such as the use of the MAP-Elites algorithm to learn the locomotion of a robot (Cully et al., 2015) can be seen as a Goal Exploration Process (GEP) where goals built with expert knowledge are provided to the algorithm.

All those frameworks leverage the common principle that in order to avoid local optima and find advanced behaviors or phenotypes, enough time should be allocated to the continued exploration of non-optimal solutions, as interesting unexpected stepping-stones could be discovered in the process and built upon afterwards, assuming a description space expressive enough to capture them.

The IMGEP framework can be applied to diverse exploration/optimization problems, and is most useful when the stepping-stones or the targets are unknown to the expert user or too complex such that they can't easily be represented and optimized as a fitness function. In that case the use of intrinsic motivations for the exploration of goals can help discover a diverse set of solutions. The use of intrinsic rewards such as based on the monitoring of the learning progress in achieving goals can further improve the efficiency of exploration by focusing on the most interesting problems and avoiding the ones that bring no more information.

A central application of this framework is the modeling of intrinsic motivations in human learning. For instance, babies show goal-directed behaviors with objects as early as 3-month old (Willatts, 1990). In chapter 3, we studied the behaviors and motivations of 21-month-old babies in a tool-use experiment, and showed that they are intrinsically motivated to explore diverse goals which often do not coincide with the target goal expected and made salient by the experimenter. In chapter 4, we investigated how the particular implementations of intrinsic motivations to self-generate goals and the representation of goals can play a role in the development of tool-use skills in a robotic model. Of course, infants learn also a lot from their parents in different ways. In chapter 5, we studied how a combination of intrinsically motivated exploration of a robotic vocal tract and of the imitation of the sounds produced by a

caregiver, that served as goals for imitation, could enable the development of speech grounded in a naturalistic play scenario with a caregiver. It would be interesting to study other ways to combine intrinsic motivations with external guidance in the context of goal exploration, as caregivers could also guide learning towards useful stepping-stones through direct input and feedback.

IMGEP has also been applied to the exploration of very different setups in other scientific domains. In Grizou et al. (2019), an IMGEP implementation allowed to discover a variety of droplet behaviors in a chemical system of self-propelling oil droplets in water, where the exploration parameters were the concentrations of the different components of the oil droplet among others. In yet another domain, Reinke et al. (2019) showed that the IMGEP framework with a goal representation learned online could find self-organized patterns in the complex morphogenetic system *Lenia*, a continuous game-of-life cellular automaton.

In the next chapter, we evaluate several IMGEP implementations in different setups such as a real tool-use robotic setup and a Minecraft tool-use environment. We show that with IMGEP, an intrinsically-motivated humanoid robot discovers a complex continuous high-dimensional environment and succeeds to explore and learn from scratch that some objects can be used as tools to act on other objects.

Chapter 8

Experimental Study of IMGEP

Summary

In this chapter, we evaluate the IMGEP architecture in several high-dimensional tool-use environments. The IMGEP architecture automatically generates a learning curriculum within several experimental setups including a real humanoid robot that can explore multiple spaces of goals with several hundred continuous dimensions. While no particular target goal is provided to the system, this curriculum allows the discovery of skills that act as stepping stone for learning more complex skills, e.g. nested tool use. We show that learning diverse spaces of goals with intrinsic motivations is more efficient for learning complex skills than only trying to learn these skills (Forestier et al., 2017).

In the previous chapter, we presented the Intrinsically Motivated Goal Exploration Processes (IMGEP) architecture for the autonomous exploration of an environment driven by self-generated goals and intrinsic rewards. In chapter 6, we obtained results in simulation showing that a modular representation of goals improves the efficiency of exploration, and that the use of intrinsic rewards based on the learning progress in each goal space further improves exploration.

Here, we evaluate the IMGEP architecture in different tool-use environments with more complex and realistic settings: a real humanoid robotic setup and a Minecraft environment. These tool-use environments are interesting benchmarks for studying the emergence of a learning curriculum as the agents can discover and explore objects that can be used as tools to move other objects in complex high-dimensional settings.

We study the behaviors of agents in the different environments depending on the learning architecture and the environment properties. We investigate in particular the benefits of a modular representation of the sensory feedback with goals based on objects, and how the exploration mutations can take into account the movement of the target object. We also compare the exploration performances with control conditions where the goals or the curriculum are hand-designed by an external user.

8.1 Tool-Use Environments

We design three tool-use environments. The first one is similar to the 2D simulated environment of chapter 6: a robotic arm with 3 joints and a gripper that can grab sticks and move toys. It is a simple environment with no physics and only 2D geometric shapes so very fast to execute. The second environment is a Minecraft scene where an agent is able to move, grab and use tools such as a pickaxe to break blocks. The third one is a real robotic setup with a Torso robot moving its arm that can reach joysticks controlling a toy robot. This setup has complex high-dimensional motor and sensory spaces with noise both in the robot physical arm and in the interaction between objects such as its hand and the joysticks. It is a high-dimensional and noisy environment with a similar stepping-stone structure as the robotic environments but with a completely different sensorimotor setup. The code of the different environments and experiments is available on GitHub¹.

8.1.1 2D Simulated Tool-Use Environment

In the 2D Simulated Environment (see Fig. 8.1), the learning agent controls a robotic arm with a gripper, that can grab one of two sticks, one with a magnet at the end and one with Velcro, that can themselves be used to move several magnets or Velcro toys. Some other objects cannot be moved, they are called static distractors, and

¹Code of the IMGEP experiments: <https://github.com/sebastien-forestier/IMGEP>

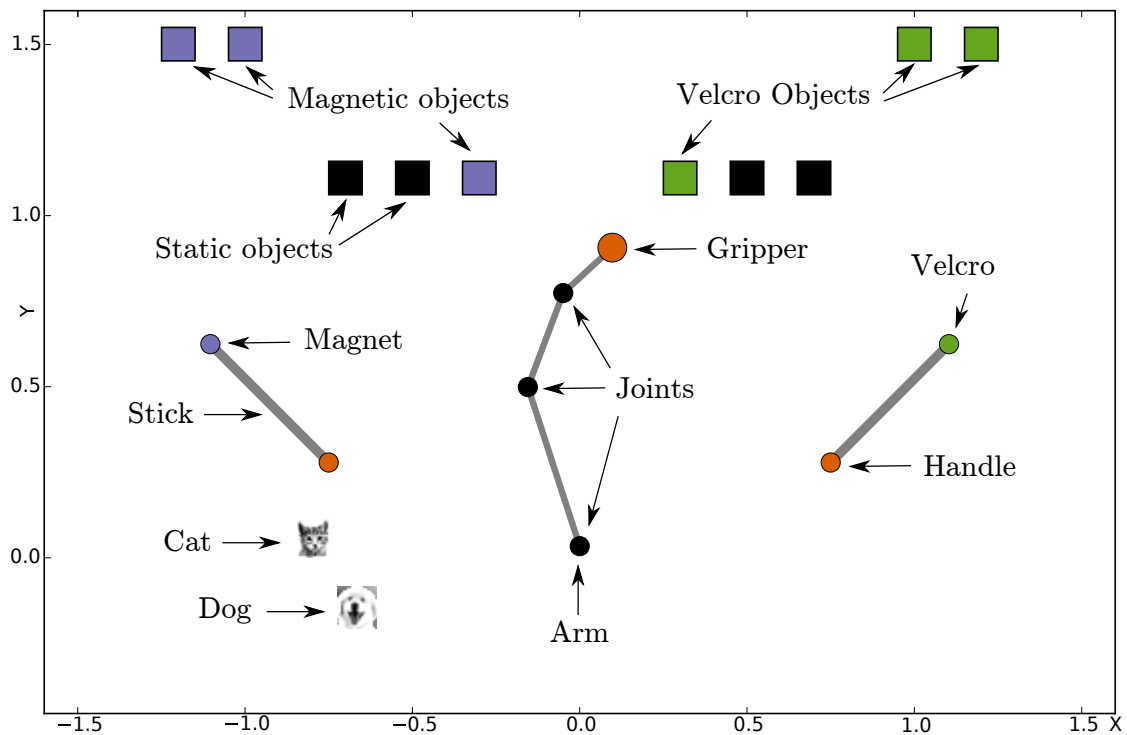


Figure 8.1: 2D Simulated Tool-Use Environment. A simulated robotic arm with a gripper can grab sticks and move toys. The gripper has to close near the handle of a stick to grab it. One magnetic toy and one Velcro toy are reachable with their corresponding stick. Other toys cannot be moved (static or too far away). The cat and the dog are distractors: they move randomly, independently of the arm.

finally a simulated cat and dog are randomly moving in the scene, they are random distractors.

The 2D robotic arm has 3 joints that can rotate from $-\pi$ rad to π rad. The length of the 3 segments of the arm are 0.5, 0.3 and 0.2 so the length of the arm is 1 unit. The starting position of the arm is vertical with joints at position 0 rad and its base is fixed at position $(0, 0)$. The gripper gr has 2 possible positions: *open* ($gr \geq 0$) and *closed* ($gr < 0$). The robotic arm has 4 degrees of freedom represented by a vector in $[-1, 1]^4$.

Two sticks of length 0.5 can be grasped by the handle side (orange side) in order to catch an out-of-reach object. The magnetic stick can catch magnetic objects (in blue), and the other stick has a Velcro tape to catch Velcro objects (in green). If the gripper closes near the handle of one stick, this stick is considered grasped and follows the gripper's position and the orientation of the arm's last segment until the gripper opens. If the other side of a stick reaches a matching object (magnetic or Velcro), the object then follows the stick. There are three magnetic objects and three

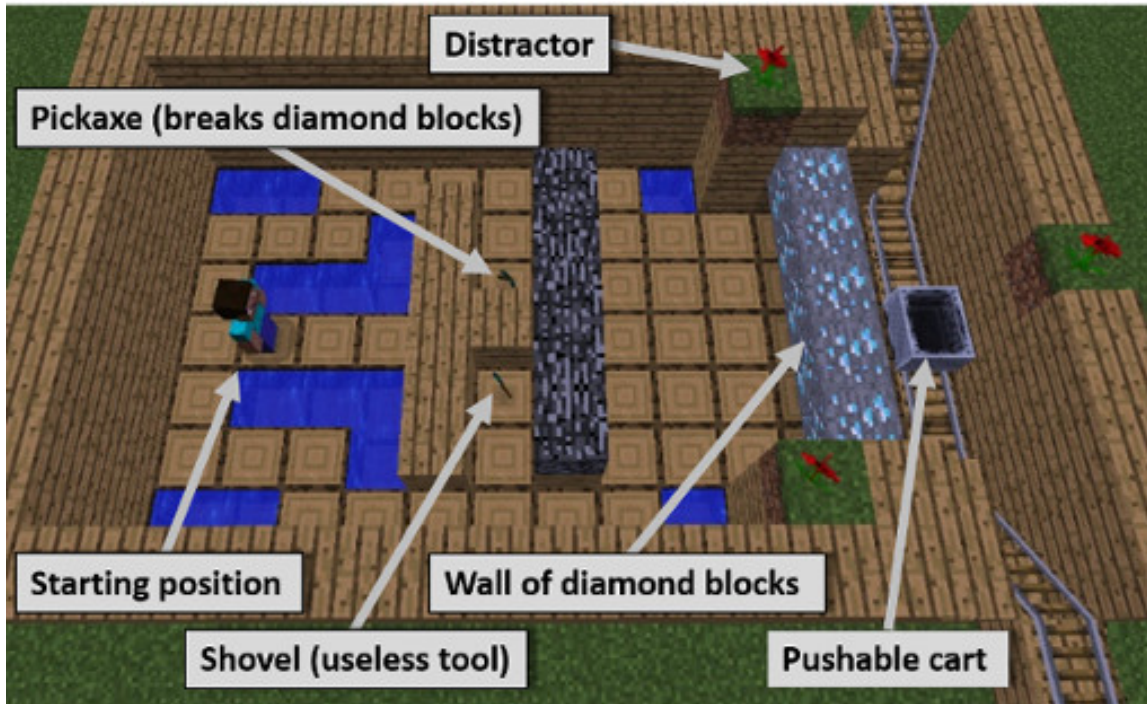


Figure 8.2: Minecraft Mountain Cart Environment created with Rémy Portelas. If the agent manages to avoid falling into water holes it may retrieve and use a pickaxe to break diamond blocks and access the cart. A shovel is also located in the arena and serves as a controllable distractor.

Velcro objects, but only one of each type is reachable with its stick. A simulated cat and dog are following a random walk, they have no interaction with the arm nor with other object. Finally, four static black squares have also no interaction with other objects. The arm, tools and other objects are reset to their initial state at the end of each iteration of 50 steps.

The agent receives a sensory feedback representing the result of its actions. This feedback (or outcome) is either composed of the position of each object at 5 time points during the 50 steps trajectory, or just the end state of each object, depending on the experiments. First, the hand is represented by its X and Y position and the aperture of the gripper (1 or -1). The sticks are represented by the X and Y position of their tip. Similarly, each other object is represented by their X and Y positions. Each of the 15 objects defines a sensory space S_i . The total sensory space S has either 155 dimensions if trajectories are represented, or 31 dimensions if only the end state of each object is represented.

8.1.2 Minecraft Mountain Cart

The Minecraft Mountain Cart (MMC) extends the famous Mountain Car control benchmark in a 3D environment with a multi-goal setting (see Fig. 8.2). This environment has been created with Rémy Portelas who I co-supervised, during his internship and beginning of PhD.

In this episodic task, the agent starts on the left of the rectangular arena and is given ten seconds (40 steps) to act on the environment using 2 continuous commands: *move* and *strafe*, both using values in $[-1; 1]$. *move(1)* moves the agent forward at full speed, *move(-0.1)* moves the agent slowly backward, etc. Similarly *strafe(1)* moves the agent left at full speed and *strafe(-0.1)* moves it slowly to the right. Additionally, a third binary action allows the agent to use the currently handled tool.

The first challenge of this environment is to learn how to navigate within the arena’s boundaries without falling in water holes (from which the agent cannot get out). Proper navigation might lead the agent to discover one of the two tools of the environment: a shovel and a pickaxe. The former is of no use but the latter enables to break diamond blocks located further ahead in the arena. A last possible interaction is for the agent to get close enough to the cart to move it along its railroad. If given enough speed, the cart is able to climb the left or right slope. The height and width of these slopes were made in such a way that an agent simply hitting the cart at full speed will not provide enough inertia for the cart to climb the slope. The agent must at least partially support the cart along the track to propel it fast enough to fully climb the slope.

The outcome of an episode is a vector composed of the end position of the agent (2D), shovel (2D), pickaxe (2D), cart (1D) and 3 distractors (2D each) positions along with a binary vector (5D) encoding the 5 diamond blocks’ states.

This environment is interesting to study modular IMGEP approaches since it is composed of a set of linked tasks of increasing complexity. Exploring how to navigate will help to discover the tools and, eventually, will allow to break blocks and move the cart.

8.1.3 Robotic Tool-Use Environment

In order to benchmark different learning algorithms in a realistic robotic environment with high-dimensional action and outcome spaces, we designed a real robotic setup composed of a humanoid arm in front of joysticks that can be used as tools to act on other objects (see Fig. 8.3). We recorded a video of an early version of the experimental setup².

A Poppy Torso robot (the learning agent) is mounted in front of two joysticks and explores with its left arm. A Poppy Ergo robot (seen as a robotic toy) is controlled by the right joystick and can push a ball that controls some lights and sounds. Poppy

²Early version of the experimental setup: <https://youtu.be/NOLAwD4ZTW0>

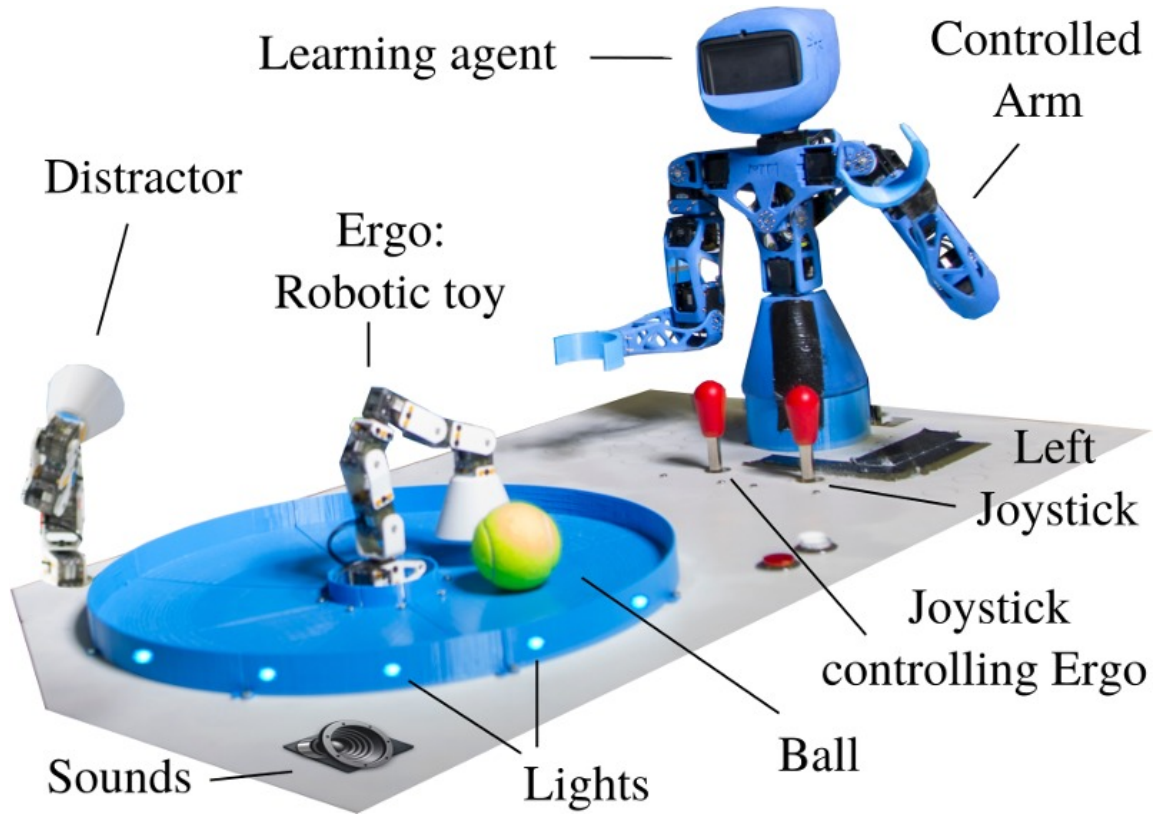


Figure 8.3: Robotic Tool-Use Environment. A Poppy Torso robot (the learning agent) is mounted in front of two joysticks that can be used as tools to act on other objects: a Poppy Ergo robotic toy and a ball that can produce light and sound.

is a robust and accessible open-source 3D printed robotic platform (Lapeyre et al., 2014).

The left arm has 4 joints, with a hook at the tip of the arm. A trajectory of the arm is here generated by radial basis functions with 5 parameters on each of the 4 degrees of freedom (20 parameters in total).

Two analogical joysticks (Ultrastick 360) can be reached by the left arm and moved in any direction. The right joystick controls the Poppy Ergo robotic toy, and the left joystick do not control any object. The Poppy Ergo robot has 6 motors, and moves with hardwired synergies that allow control of rotational speed and radial extension. A tennis ball is freely moving in the blue arena which is slightly sloped so that the ball comes close to the center at the end of a movement. The speed of the ball controls (above a threshold) the intensity of the light of a LED circle around the arena. Finally, when the ball touches the border of the arena, a sound is produced and varied in pitch depending on ball position.

Several other objects are included in the environment, with which the agent cannot interact. Two simulated 2D objects are moving randomly, independently of the agent

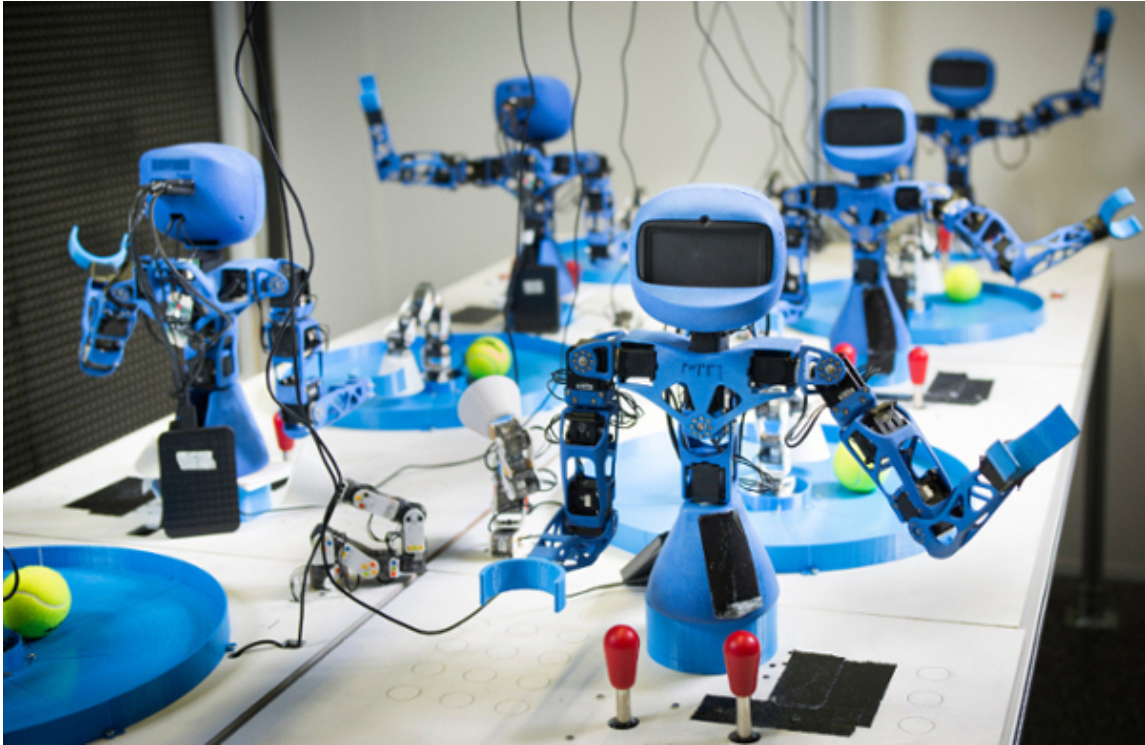


Figure 8.4: Six copies of the setup are running in parallel to gather more data. Some Ergo robots are placed between robots: they act as distractors that move randomly, independently of the agents.

(imagine a cat and a dog playing together), with a random walk. Six objects are static: the right hand (3D) of the robot that is disabled in this experiment, the camera recording the ball trajectory (3D), the blue circular arena (2D), an out-of-reach yellow toy (2D), the red button also out-of-reach (2D) and the lamp (2D). All distractor objects are reset after each roll-out.

The context c of this environment represents the current configuration of objects in the scene. In practice, since only the Ergo and ball are not reset after each roll-out, this amounts to measuring the rotation angle of the Ergo and of the ball around the center of the arena.

The agent receives a sensory feedback representing the result of its actions. We assume that there is a perceptual system providing the trajectories of all objects in the scene. First, the 3D trajectory of the hand is computed through a forward model of the arm as its x , y and z position. The 2D states of each joystick and of the Ergo are read by sensors, and the position of the ball retrieved through the camera. The states of the 1D intensity of the light and the 1D pitch of the sound are computed from the ball position and speed. Each of the 15 objects defines a sensory space S_i representing its trajectory. The total sensory space S has 310 dimensions.

8.2 Implementation of the IMGEP architecture

In the following subsections, we detail our implementation of the algorithmic parts of the modular population-based IMGEP architecture (see architecture in chapter 7).

8.2.1 Motor Policy π_θ

In the 2D Simulated environment and the Robotic environment, we implement the motor policies with Radial Basis Functions (RBF). We define 5 Gaussian basis functions with the same shape ($\sigma = 5$ for a 50 steps trajectory in the 2D environment and $\sigma = 3$ for 30 steps in the Robotic environment) and with equally spaced centers (see Fig. 8.5). The movement of each joint is the result of a weighted sum of the product of 5 parameters and the 5 basis. The total vector θ has 20 parameters, in both the 2D Simulated and the Robotic environment. In the 2D environment, the fourth joint is a gripper that is considered open if its angle is positive and closed otherwise.

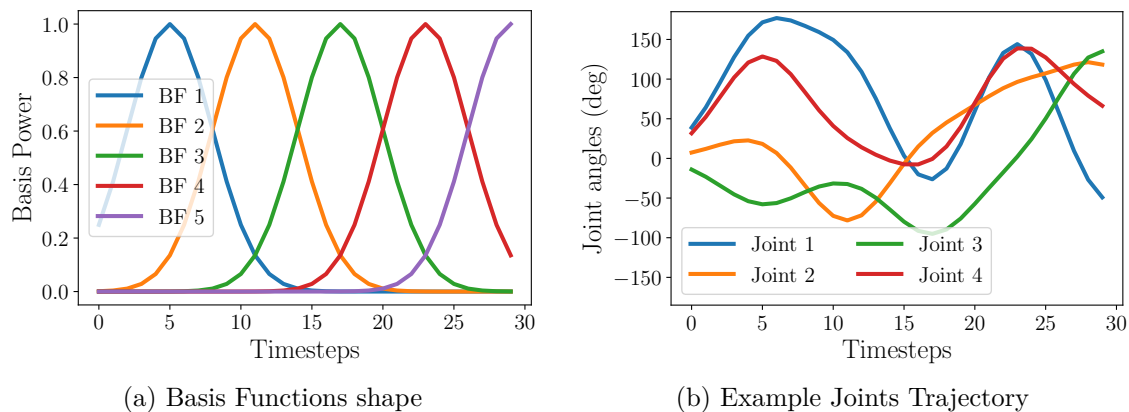


Figure 8.5: Implementation of motor policies π_θ through Radial Basis Functions. (a) 5 Gaussian bases with different centers but same shape. (b) the movement of each joint is the result of a weighted sum of the product of 5 parameters and the 5 basis. The total vector θ has 20 parameters, in both the 2D Simulated and the Robotic environment. In the 2D environment, the fourth joint is a gripper that is considered open if its angle is positive and closed otherwise.

In the Minecraft Mountain Cart environment, trajectories are sampled in a closed-loop fashion using neural networks. The observation vector has the same structure as the outcome vector: it provides the current positions of all objects normalized in $[-1; 1]$ (18D). Each neural network is composed of one hidden layer of 64 relu units and a 3D output with tanh activation functions. The 1411 policy parameters are initialized using the initialization scheme of He et al. (2015).

8.2.2 Temporal Modularity: Stepping-Stone Preserving Mutations

When targeting a new goal g , the exploration meta-policy infers the best policy π_θ to reach the goal g and performs a mutation of θ in order to explore new policies. The mutation operator could just add a random noise on the parameters θ , however, those parameters do not all have the same influence on the execution of the policy. In our implementations, the parameters are sequenced in time, with some parameters influencing the beginning of the policy roll-out and some the end of the trajectory. However, in the context of tool use, the reaching or grasping of a tool is necessary for executing a subsequent action on an object. A random mutation of policy parameters irrespective of the moment when the tool is grasped or the object is reached with the tool results in an action sequence where the agent do not reach or grasp the tool and thus cannot explore the corresponding object.

We design a Stepping-Stone Preserving Mutation operator (`SSPMutation`) that analyzes the movement features of the target object while the best motor policy π_θ was previously run, to find the moment when the object started to move. The operator does not change the variables of θ concerning the movement before the object moved and modifies the variables of θ concerning the movement after the object moved. When the goal of the agent is to move the tool and it already succeeded to move the tool in the past with policy π_θ , then the application of this mutation operator changes the behavior of the agent only when the tool starts to move, which ensures grasping the tool. Similarly, when the goal of the agent is to move a toy controlled by a tool, the mutation changes the behavior only when the toy starts to move, which makes the agent grasp the tool and reach the toy before exploring new actions, so that the agent does not miss the tool nor the toy. The idea of this stepping-stone preserving operator is similar to the Go-Explore approach (Ecoffet et al., 2019).

The Stepping-Stone Preserving Mutation operator adds a Gaussian noise around those values of θ in the 2D simulated environment ($\sigma = 0.05$) and in Minecraft Mountain Cart ($\sigma = 0.3$), or adds the Gaussian noise around the previous motor positions (in the robotic environment with joysticks). In the experimental section we compare it to the `FullMutation` operator that adds a Gaussian noise to θ irrespective of the moment when the target object moved.

8.2.3 Goal Space Policy Γ

The agent estimates its learning progress globally in each goal space (or for each model learned). At each iteration, the context c is observed, a goal space k is chosen by Γ and a random goal g is sampled by γ_k in \mathcal{G}^k (corresponding to a fitness function f_g). Then, in 80% of the iterations, the agent uses $\Pi_\epsilon(\theta \mid g, c)$ to generate with exploration a policy θ and does not update its progress estimation. In the other 20%, it uses

Π , without exploration, to generate θ and updates its learning progress estimation in \mathcal{G}^k , with the estimated progress in reaching g . To estimate the learning progress r_i made to reach the current goal g , the agent compares the outcome o_τ with the outcome o'_τ obtained for the previous context and goal (g', c' most similar (Euclidean distance) to (g, c)): $r_i = f_g(\tau) - f_g(\tau')$. Finally, Γ implements a non-stationary bandit algorithm to sample goal spaces. The bandit keeps track of a running average r_i^k of the intrinsic rewards r_i associated to the current goal space \mathcal{P}^k . With probability 20%, it samples a random space \mathcal{P}^k , and with probability 80%, the probability to sample \mathcal{P}^k is proportional to r_i^k in the 2D Simulated and Minecraft environments, or $\exp(\frac{r_i^k}{\sum_k(r_i^k)})$ if $r_i^k > 0$ and 0 otherwise, in the Robotic environment.

8.2.4 Control Conditions

We design several control conditions to understand the influence of each component of the IMGEP architecture. In the Random Model Babbling (RMB) condition, the choice of goal space (or model to train) is random: $\Gamma(\mathbf{k} \mid \mathbf{c})$, and $\gamma_k(\mathbf{g} \mid \mathbf{c})$ for each k are always uniform distributions. Agents in the Single Goal Space (SGS) condition always choose the same goal space, of high interest to the engineer: the magnet toy in the 2D Simulated environment, and the ball in the robotic environment. The Fixed Curriculum (FC) condition defines Γ as a curriculum sequence engineered by hand: the agents explore objects in a sequence from the easiest to discover to the most complex object while ignoring distractors. The conditions SGS and FC are thus extrinsically motivated controls. We define the Flat Random Goal Babbling (FRGB) condition with a single outcome/goal space containing all the variables of all objects, to compare modular and non-modular representations of the environment. The agents in this condition choose random goals in this space, and use the `FullMutation` operator. Finally, agents in the Random condition always choose random motor policies θ .

8.3 Results

In this section we show the results of several experiments with the three environments and the different learning conditions. We first study in details the Active Model Babbling (AMB) learning algorithm, a modular implementation of the IMGEP architecture. Then, in order to understand the contribution of the different components of this learning algorithm, we compare it to several controls (or ablations): without a modular representation of goals, without the goal sampling based on learning progress, or without the stepping-stone preserving mutation operator. In those experiments, goals are sampled in spaces representing the sensory feedback from the environment. We thus compare several possible encodings of the feedback: with the trajectory of each object or with only the end point of the trajectories. We included distractors

		Rdm	SGS	Flat	RMB	AMB	FC
2D Simu Env	M. Tool	0,0,0	0,0,0	8,0,11,13	33,36,39	57, 61 ,65	61, 67 ,70
	M. Toy	0,0,0	0,0,0	0,0,0	0,0,5.0	0, 3.0 ,16	0, 3.0 ,19
Minecraft Mountain Cart	XY	28,29,30	29,29,30	34,36,40	48,50,54	55,58,61	59, 63 ,67
	Shovel	5,5,6	5,6,7	8,11,13	25,27,30	32,34,37	34, 37 ,42
	Pickaxe	6,6,7	6,7,8	11,15,19	33,35,39	41,45,48	43, 51 ,61
	Blocks	3,3,3	3,3,3	3,11,19	69,77,84	73, 84 ,93	100, 100 ,100
	Cart	0,0,0	0,0,0	0,0,1	5,162,409	56,360,886	386, 787 ,1207
Robotic Env	Hand	24,24,25	18,19,20	20,21,22	22,24,25	22,23,24	21,22,23
	L. Joy.	4.2,4.7,5.9	1.9,3.3,4.6	0.1,0.1,0.3	15,18,19	20, 22 ,26	23, 26 ,29
	R. Joy.	0.6,0.9,1.0	0.3,0.4,0.5	0,0,0	10,11,13	16, 18 ,22	15,17,18
	Ergo	0.2,0.3,0.4	0.1,0.1,0.2	0,0,0	1.2, 1.5 ,1.7	1.5, 1.7 ,1.8	1.7, 1.7 ,1.9
	Ball	0,0,0.1	0,0,0	0,0,0	0.8,1.0,1.0	0.9, 1.1 ,1.2	0.9, 0.9 ,1.0
	Light	0.1,0.1,0.1	0.1,0.2,0.2	0.1,0.1,0.1	0.8,1.8,3.0	2.0, 3.6 ,4.9	1.8, 2.2 ,3.7
	Sound	0.1,0.1,0.1	0.1,0.1,0.1	0.1,0.1,0.1	0.8,1.1,2.6	1.7, 2.8 ,3.6	1.2,1.6,2.3

Table 8.1: Summary of the exploration result at the end of the runs, in all conditions in all spaces of all environments. We give the 25, 50 and 75 percentiles of the exploration result of all seeds. The exploration measures the percentage of reached cells in a discretization of each goal space. The best condition in each space is highlighted in bold, based on Welch’s t-tests (with threshold $p < 0.05$): if several conditions are not significantly different, they are all highlighted. In the 2D Simulated environment, there are 100 seeds for each condition, and the exploration measures the number of cells reached in a discretization of the 2D space of the end position of each object with 100 bins on each dimension. In the Minecraft environment, there are 20 runs with different seeds for condition Random, SGS, FRGB, FC and 42 for AMB and RMB. The exploration metric for the agent, pickaxe and shovel spaces is the number of reached cells in a discretization of the 2D space in 450 bins (15 on the x axis, 30 on the y axis). The same measure is used for the block space, which is already discrete and has 32 possible combinations. For the cart space we measure exploration as the number of different outcomes reached. In the Robotic environment, there are 6 runs with different seeds for condition SGS, 8 for FRGB, 16 for RMB, 23 for AMB, 12 for FC and 6 for Random, and the exploration also measures the number of cells reached in a discretization of the space of the end position of each object with 1000 bins in 1D, 100 bins on each dimension in 2D, and 20 bins on each dimension in 3D.

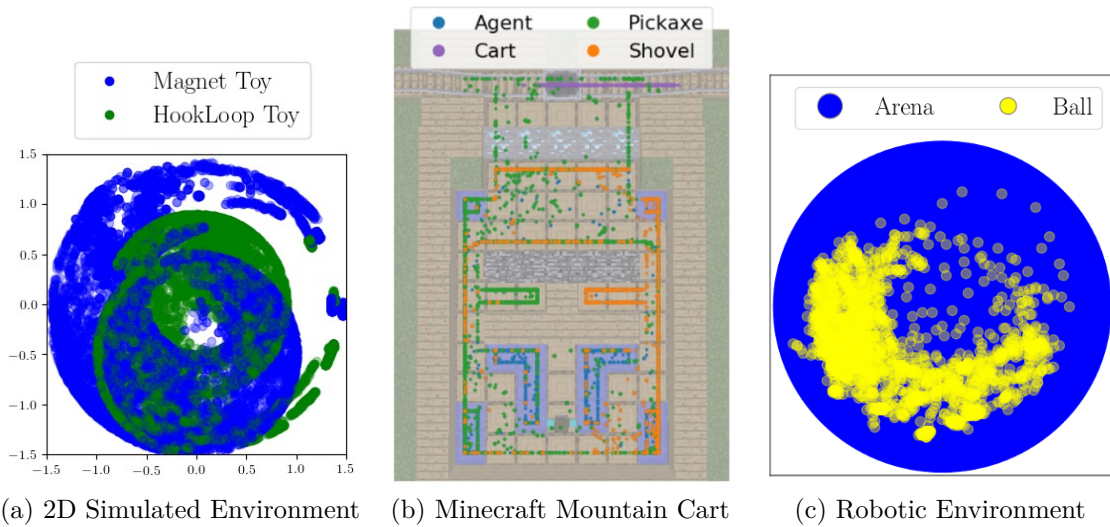


Figure 8.6: Examples of exploration map of one IMGEP agent in each environment. (a) in the 2D Simulated Environment, we plot the position of the reachable magnet toy at the end of each iteration with a blue point, and the Velcro toy in green. (b) in Minecraft Mountain Cart we plot the end position of the agent, the agent with pickaxe, the agent with shovel, and the cart. (c) in the Robotic environment, the position of the ball is plotted when it moved in the arena.

that cannot be controlled by the learning agent in the three tool-use environments. We also test the learning conditions with and without distractors to evaluate their robustness to distractors. A summary of the exploration efficiency of all agents in all environments is provided in Table 8.1 together with additional details.

8.3.1 Intrinsically Motivated Goal Exploration

Here we study in detail the Active Model Babbling (AMB) learning algorithm. AMB agents encode the sensory feedback about objects with a modular representation: each object is associated with one independent learning module. At each iteration, they first select an object to explore, then a particular goal to reach for this object. They execute a motor policy to reach this goal, and observe the outcome. The selection of the object to explore is based on a self-estimation of the learning progress made to move each object according to chosen goals. The AMB algorithm is thus a modular implementation of the IMGEP architecture.

Exploration Maps

We first plot examples of exploration results as cumulative exploration maps, one per environment. Those maps show all the positions where one AMB agent succeeded to

move objects.

Fig. 8.6(a) shows the position of the reachable toys of the 2D simulated environment at the end of each iteration in one trial of intrinsically motivated goal exploration. The reachable area for those two toys is the inside the circle of radius 1.5 and center 0. We can see that in 100k iterations, the agent succeeded to transport the toys in many places in this area. The experiments with other seeds are very similar. Fig. 8.6(b) shows an exploration map of a typical run in Minecraft Mountain Cart after 40k iterations. As you can see the agent successfully managed to (1) navigate within the arena boundaries, (2) move the pickaxe and shovel, (3) use the pickaxe to break blocks and (4) move the cart located behind these blocks. An example in the robotic environment is shown in Fig. 8.6(c) where we plot the position of the ball when it moved in the first 10k iterations of the exploration of one agent.

Overall, they show that IMGEP agents discovered how to use the different tools in each environment within the time limit: the sticks to grab the toys in the 2D simulated environment, the pickaxe to mine blocks to reach the cart in Minecraft Mountain Cart, the joysticks to move the toy and push the ball in the robotic experiment.

Discoveries

In order to understand the tool-use structure of the exploration problem in each environment, we can look in more details how agents succeeded to move objects while exploring other objects. Indeed, to the agents starting to explore, tools are objects like any other object (e.g. the hand, the stick and the ball have the same status). However, if a tool needs to be used to move another object, then this tool will be discovered before that object, so the exploration of this tool is a stepping-stone giving more chances to discover novelty with that object than the exploration of any other object. To quantify these dependencies between objects in our tool-use environments, we show the proportion of movements where an object of interest has been moved depending on the currently explored object.

Concerning the 2D simulated environment, Fig. 8.7 shows the proportion of the iterations with a goal in a given space that allowed to move (a) the magnet tool, (b) the magnet toy, in 10 runs with different seeds. First, random movements of the arm have almost zero chances to reach the magnet tool or toy. Exploring movements of the hand however have about 1.5% chances to move the magnet tool, but still almost zero chances to reach the toy. Exploring the magnet tool makes this tool move in about 93% of the iterations, and makes the toy move in about 0.1% of movements. Finally, exploring the toy makes the tool and the toy move with a high probability as soon as the toy was discovered. Those results illustrate the stepping-stone structure of this environment, where each object must be well explored in order to discover the next step in complexity (Hand \rightarrow Tool \rightarrow Toy).

In Minecraft Mountain Cart (see Fig. 8.7(c,d,e)), random exploration with neural networks in this environment is extremely challenging. An agent following random

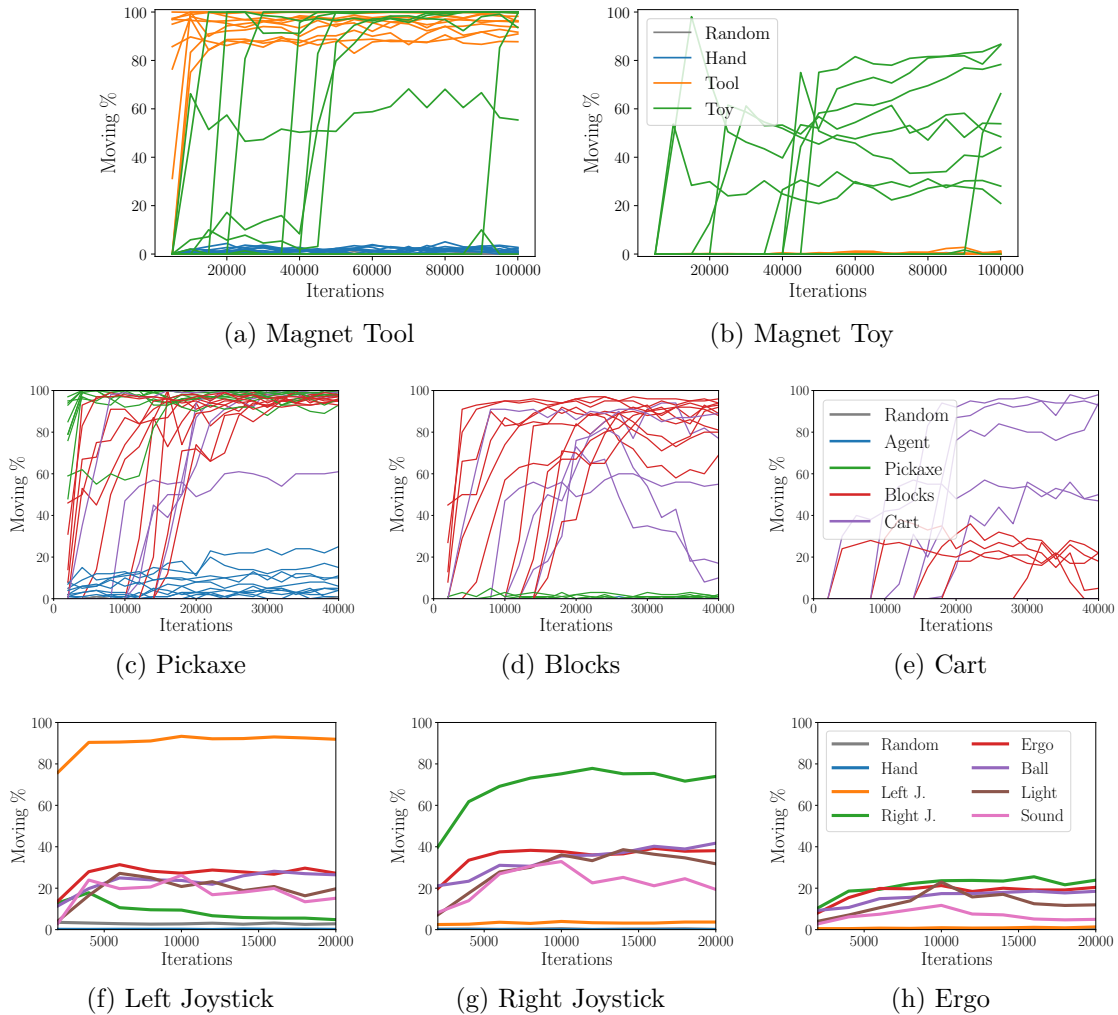


Figure 8.7: Stepping-stone structure of the three environments. In the 2D Simulated environment, we show the proportion of iterations that allowed to (a) move the magnet tool, (b) move the magnet toy, depending on the currently explored goal space (or random movements), for 10 IMGEP agents. The fastest way to discover the tool is to explore the hand and to discover the toy is to explore the tool. In the Minecraft Mountain Cart environment, we show the proportion of iterations that allowed to (c) move the pickaxe, (d) mine diamond blocks, and (e) move the cart, depending on the currently explored goal space (or random movements), for 10 agents with different seeds. Exploring the agent space helps discover the pickaxe, exploring the pickaxe helps discover the blocks, and exploring the blocks helps discover the cart. In the Robotic environment, we show the proportion of iterations that allowed to (f) reach the left joystick, (g) reach the right joystick, and (h) move the Ergo robot, depending on the currently explored goal space (or random movements), averaged for 11 IMGEP agents with different seeds. Exploring random movements or the Hand space helps discover the left joystick, exploring the left joystick helps discover the right one, which helps discover the Ergo toy.

policies has 0.04% chances to discover the pickaxe, 0.00025% chances to break a single block and it never managed to move the cart (over 800k episodes). IMGEP agents reach better performances by leveraging the sequential nature of the environment: when exploring the agent space there is around 10% chances to discover the pickaxe, and exploring the pickaxe space has around 1% chances to break blocks. Finally, exploring the block space has about 8% chances to lead an agent to discover the cart.

In the Robotic environment, a similar stepping-stone exploration structure is displayed (see Fig. 8.7(f,g,h)): in order to discover the left joystick, the robots needs to do random movements with its arm, which have about 2.9% chances to makes the left joystick move, or explore its hand (0.2% chance). To discover the right joystick, the agent has to explore the left joystick, which gives a probability of 3.3% to reach the right one. To discover the Ergo (the white robotic toy in the center of the blue arena), the exploration of the right joystick gives 23% chances to move it, whereas the exploration of the Hand, the left joystick or random movements has a very low probability to make it move.

Learned Skills

In Minecraft Mountain Cart we performed post-training tests of competence in addition of exploration measures. Using modular approaches allows to easily test competence on specific objects of the environment. Fig. 8.8(b) shows an example in the cart space for an AMB agent. This agent successfully learned to move the cart close to the 5 queried locations.

For each of the RMB, AMB and FC runs we performed a statistical analysis of competence in the cart and pickaxe spaces using 1000 and 800 uniformly generated goals, respectively. We were also able to test SGS trials for cart competence as this condition has the cart as goal space. A goal is considered reached if the Euclidean distance between the outcome and the goal is lower than 0.05 in the normalized space (in range $[-1, 1]$) for each object. Since the pickaxe goal space is loosely defined as a rectangular area around the environment’s arena, many goals are not reachable. Results are shown in Table 8.2. SGS agents never managed to move the cart for any of the given goals. AMB appears to be significantly better than RMB on the pickaxe space ($p < 0.01$ on Welch’s t-tests). However it is not in the cart space ($p = 0.09$), which might be due to the stochasticity of the environment. FC is not significantly better than AMB on the cart and pickaxe spaces.

Intrinsic Rewards based on Learning Progress

The IMGEP agents self-evaluate their learning progress to control each object. When they choose a goal for an object, they monitor what is the actual movement given to the object and compare it to the goal. If the distance between the goals and the actual reached movements decrease over time on average, this tells the agents it is

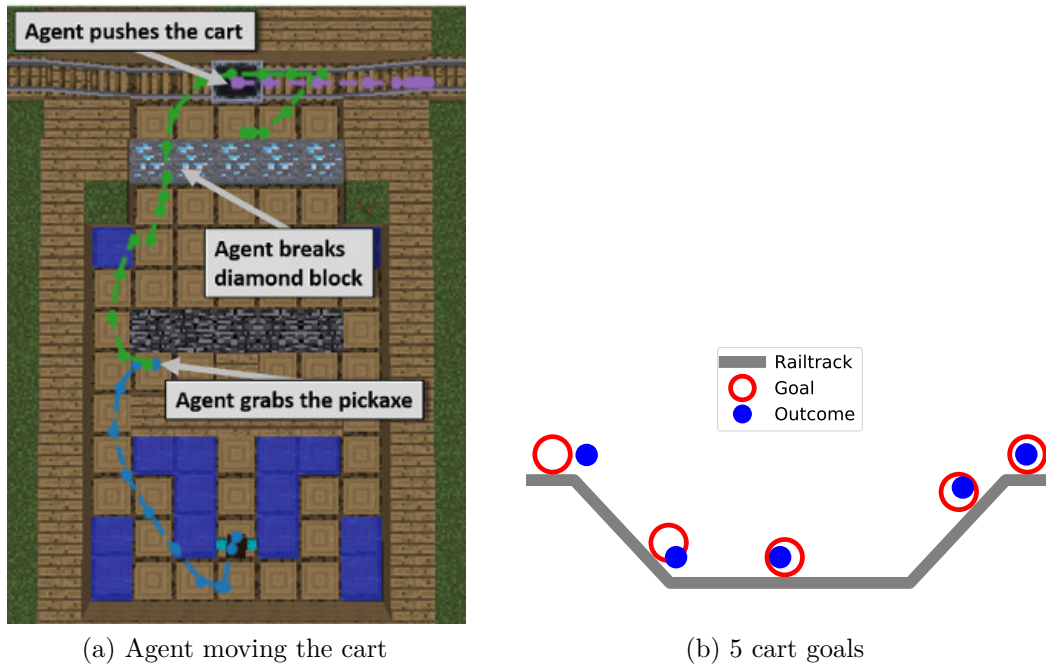


Figure 8.8: Example of learned skills in the Minecraft Mountain Cart. (a) One AMB agent’s trajectory for a single cart goal. (b) Five final cart positions reached by an AMB agent when tasked to reach five different targets. This agent successfully learned to push the cart along the track.

making progress to control this object. This signal is used as an intrinsic reward signal that the agent will seek to maximize by choosing to explore objects that yield a high learning progress. We can analyze this signal to understand at which point the agent perceived progress to control each object and how its exploration behavior changed over time.

Fig. 8.9 (top) shows the intrinsic rewards of two agents (different seeds) to explore each object in the 2D simulated environment, computed by the agents as the average of intrinsic rewards based on learning progress to move each object. We can see that the intrinsic reward of the hand increases first as it is the easiest object to move. Then, when the sticks are discovered, the agents start to make progress to move them in many directions. Similarly, while exploring the sticks, they discover the reachable toys, so they start making progress in moving those toys. However, the static objects can’t be moved so their learning progress is strictly zero, and the objects moving randomly independently of the agent (cat and dog) have a very low progress.

Fig. 8.9 (middle) shows the intrinsic reward of two agents in the Minecraft Mountain Cart environment. Both agents first explore the simpler agent space and then quickly improves on the shovel and pickaxe spaces. Exploring the pickaxe space leads to discover how to progress in the block space. Finally, after some progress in the block

	Pickaxe goals	Cart goals
FC	39,49,55	12,17,25
AMB	41,45,49	8,11,18
RMB	37,40,43	6,9,15
SGS	N/A	0,0,0

Table 8.2: Competence results in Minecraft Mountain Cart. We give the 25, 50 and 75 percentiles of the competence result of all seeds.

space, the cart is discovered after 14k episodes for the first agent (left figure) and 26k episodes for the other (right figure). The 3 distracting flowers have an interest strictly equal to zero in both runs.

Fig. 8.9 (bottom) shows the intrinsic reward of two agents in the Robotic environment. The first interesting object is the robot’s hand, followed by the left joystick and then the right joystick. The left joystick is the easiest to reach and move so it gets interesting before the right one in most runs, but then they have similar learning progress curves. However, the right joystick can be used as a tool to control other objects, so that one will be touched more often. Then, the agent can discover the Ergo and Ball while exploring the joysticks. Finally, some agents also discover that the ball can be used to make light or sound. Here also, the progress of static objects is zero and the one of random objects is low.

Overall, the evolution of those interests show that evaluating the learning progress to move objects allows agents to self-organize a learning curriculum focusing on the objects currently yielding the most progress and to discover stepping stones one after the other.

8.3.2 Influence of Goal Modularity

In this section, we study several algorithms with a different goal space structure in order to evaluate the influence of the modularity of the goal space. We compare the Active Model Babbling condition to other conditions. In the Flat Random Goal Babbling (FRGB) condition, the goal space is not modular and contains all variables of all objects. With this non-modular sensory representation, agents choose goals in the whole sensory space, which corresponds to all objects: a goal could mean for instance push *toy1* to the left and *toy2* to the right at the same time, which might be unfeasible. This exploration dynamics results in exploring the most diverse global sensory states, which is akin to novelty search algorithms (Lehman and Stanley, 2011a). We also test the Random control where agents always choose random actions.

In the 2D simulated environment, we run 100 trials of each condition with different random seeds. We measure the exploration of one stick and its corresponding toy as the cumulative number of reached cells in a discretization of the 2D space of the

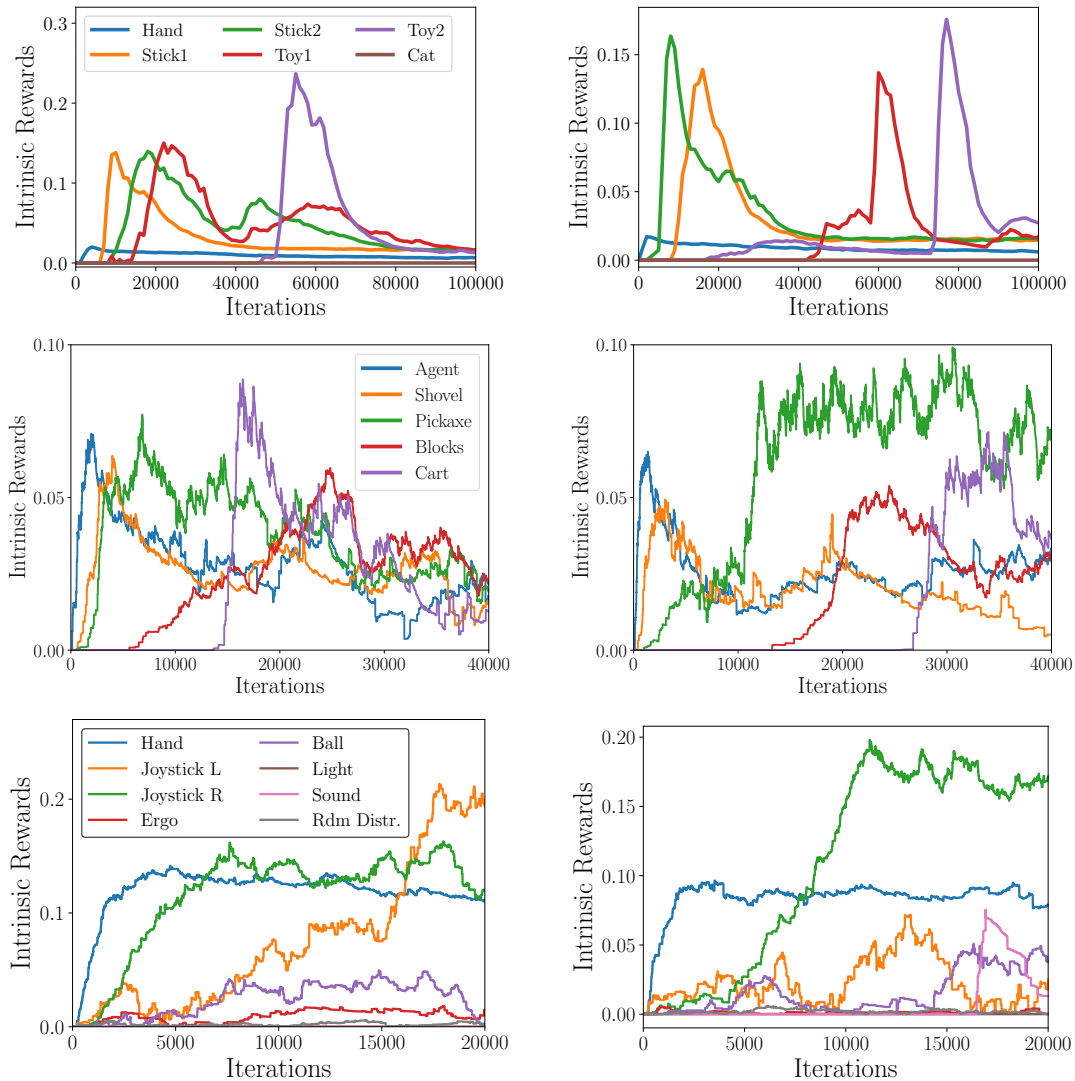


Figure 8.9: Examples of intrinsic rewards in the three environments. In the 2D Simulated environment (top), agents are first interested in exploring their hand as this is the only object they manage to move, until they discover one of the sticks. Then they make progress to move the stick, so the intrinsic reward for moving the stick increases and they focus on it, and then on the other objects they discover: the other stick and the two toys. They make no progress to move the distractors so those intrinsic reward are always zero. In the Robotic environment (middle), agents first succeed to move their hand, so they focus on this object at the beginning, until they discover the joysticks. The exploration of the right joystick makes them discover the Ergo toy, which can push the Ball. Some agents also discover how to produce light and sound with the Ball. Agents have a low intrinsic reward for exploring random distractors. In Minecraft Mountain Cart (bottom), agents first focus on exploring the space of their position until they discover the shovel or the pickaxe and start making progress to move them. When they discover how to mine blocks with the pickaxe and to push the cart, they make progress in those goal spaces, get intrinsic rewards and thus focus more on these.

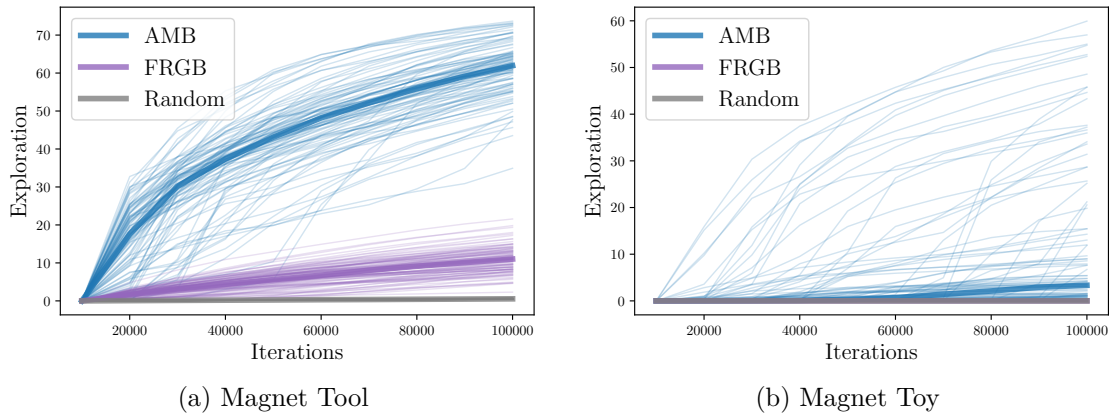


Figure 8.10: Influence of Goal Modularity on exploration in the 2D simulated environment. The agents using a modular representation (Active Model Babbling) explore much better the tool and toy spaces than agents with a flat representation (Flat Random Goal Babbling). Control agents always choosing completely random actions do not manage to touch a toy with the stick.

position of each objects at the end of movements. Fig. 8.10 shows the evolution of the exploration of the stick and the toy in the 100 trials of each condition. We plot in bold the median over the 100 trials in each condition. We can see that the modularity of the goal space helps exploration: the median exploration after 100k iterations is about 60% of the magnet tool space for condition AMB vs about 10% for condition FRGB. The agents in condition AMB succeeded to reach the magnet toy, with a substantial variance between the 100 trials. Some AMB agents explored very well the magnet toy (up to 60%) and some did not (very low exploration). Finally, completely random agents did not even manage to explore the magnet tool.

Fig. 8.12 shows exploration results in the Minecraft Mountain Cart environment for 20 trials of all conditions except for AMB and RMB which were run 42 times. When looking at the median exploration in the pickaxe space, FRGB does not manage to reach more than 15% exploration when AMB and RMB reached 45% and 35%, respectively. Modular approaches significantly outperform FRGB across all goal spaces (Welch's t-tests at 40k iterations, $p < 0.001$). Random agents did not manage to explore the block and cart spaces.

In the robotic environment (see Fig. 8.13), agents with the flat (intricate) representation of the sensory feedback (FRGB) do not explore objects other than the hand.

The modular representation of the sensory space thus greatly improves exploration efficiency compared to a flat intricate representation of the whole sensory feedback, as it allows to consider the different objects independently to monitor their behavior and select disentangled goals.

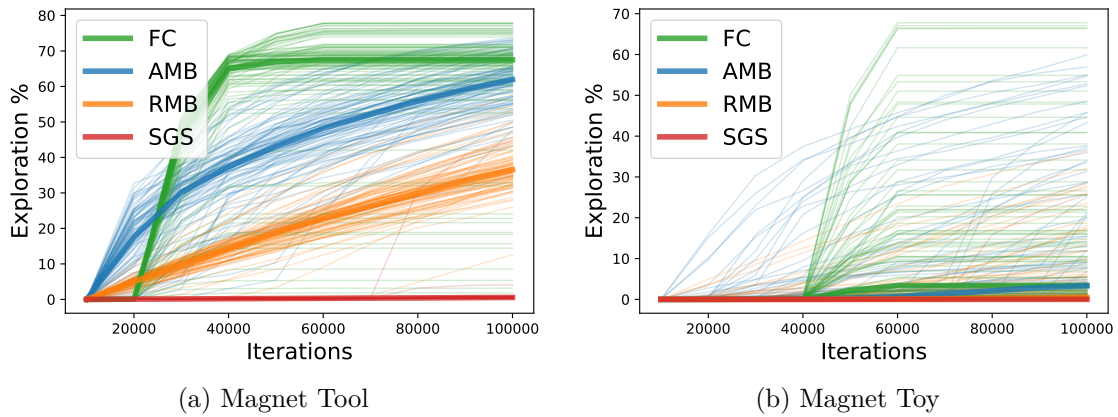


Figure 8.11: Influence of curriculum learning on exploration in the 2D Simulated environment. Agents self-organizing their curriculum (Active Model Babbling) based on their learning progress explore better than agents choosing to explore random objects (Random Model Babbling) or agents choosing always to explore the magnet toy (Single Goal Space). Agents with a hard-coded curriculum learning sequence from the simpler objects to the most complex have similar exploration results than autonomous AMB agents after 100k iterations.

8.3.3 Curriculum Learning

A modular sensory representation based on objects allows AMB agents to self-monitor their learning progress to control each object, and to accordingly explore objects with high learning progress. Here, we compare several variants of agents with a modular sensory representation based on objects, but with a different choice of object to explore. To evaluate the efficiency of the sampling based on learning progress, we define condition Random Model Babbling (RMB), where agents always choose objects to explore at random. The sampling based on learning progress of AMB agents makes agents explore any object that shows learning progress, and ignore objects that do not move, are fully predictable, or move independently of the agent. However if we are only interested in a particular complex skill that we want the agent to learn, such as moving the ball in the robotic environment, then it is not obvious if supervising learning by specifying a curriculum targeted at this skill can accelerate the learning of this skill. We thus define two control conditions with a hand-designed curriculum. In condition Single Goal Space (SGS), agents always choose goals for the same complex target object: the magnet toy in the 2D simulated environment, the cart in Minecraft Mountain Cart, or the ball in the robotic environment. In condition Fixed Curriculum (FC), a particular sequence of exploration is specified, from the easier stepping-stones to the more complex ones. In the 2D simulated environment, the agent samples goals for 20k iterations on each object in this order: hand, magnet

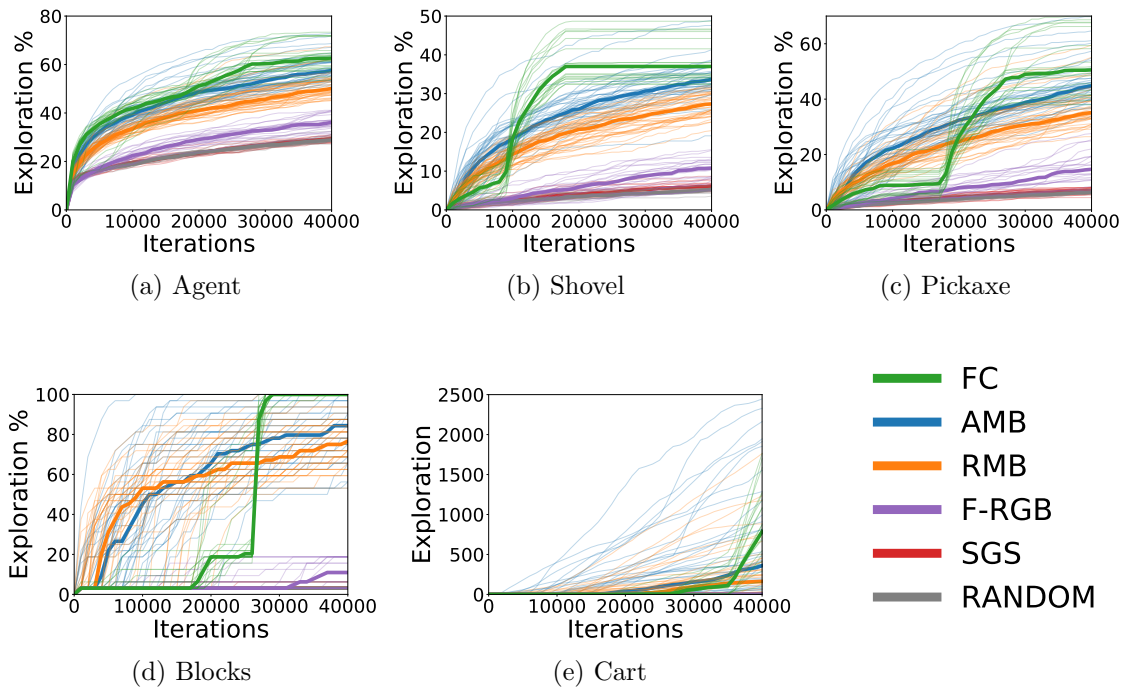


Figure 8.12: Exploration results in Minecraft Mountain Cart. Modular approaches (AMB and RMB) performs significantly better than the flat (F-RGB) approach. Agents actively generating their curriculum (AMB) perform better overall than agents choosing goal spaces randomly (RMB). Focusing on the cart space (SGS) is equivalent to performing random policies (Random). For the agent, pickaxe and shovel spaces, exploration is measured as the cumulative number of reached cells in a discretization of the 2D space. For the block and cart spaces we measure the number of unique outcomes reached.

tool, magnet toy, Velcro tool, Velcro toy. In the robotic environment, we define the sequence as the following: hand, left joystick, right joystick, ergo, ball, light and sound.

Fig. 8.11 shows the exploration evolution in the 2D simulated environment. First, we can see that the sampling based on learning progress (AMB) helps exploration of the tool and the toy compared to the random choice of object to explore (RMB): 62% vs 37% for the tool and 3.3% vs 0.5% for the toy. Agents in the SGS condition did not manage to explore the tool and the toy. Agents with a predefined curriculum succeeded to explore the tool and the toy very well, the tool between 20k and 40k iterations and the toy between 40k and 60k iterations, with a median slightly better than in AMB.

Fig. 8.12 shows the evolution of exploration in Minecraft Mountain Cart. Agents focusing their goal sampling on the cart space (SGS) have low performances across all

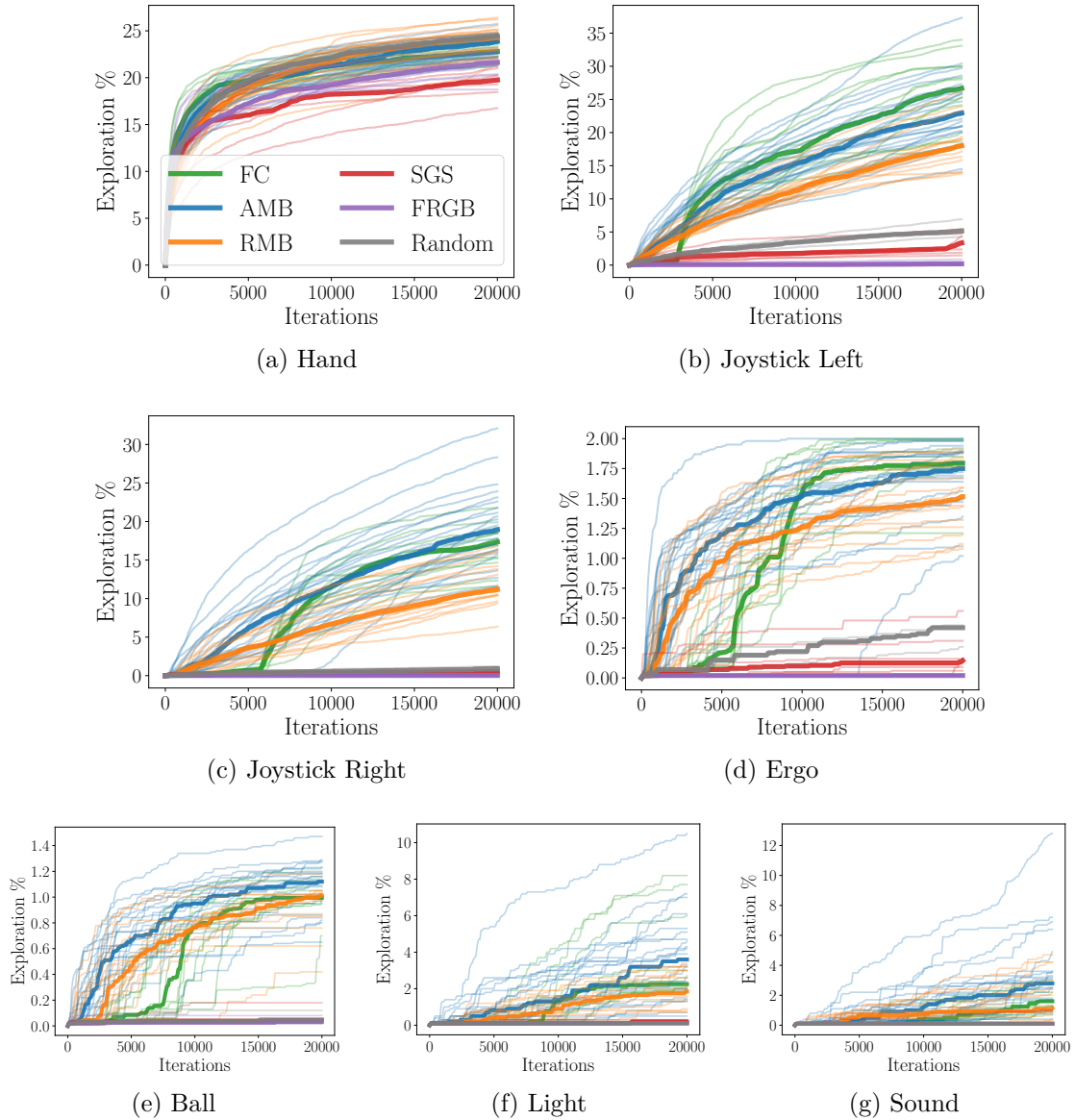


Figure 8.13: Exploration results in the Robotic environment. Agents self-organizing their curriculum (Active Model Babbling) based on their learning progress explore better than agents choosing random objects (Random Model Babbling) in the joysticks, ball, light and sound spaces, and better than agents with a hard-coded curriculum sequence (FC) in the right joystick and sound spaces. Agents always choosing to explore the ball (Single Goal Space) and agents without a modular representation of goals (FRGB) have low exploration results in all spaces.

goal spaces, especially for the cart and block spaces which are never discovered. Agents using learning progress sampling (AMB) explored significantly more than random sampling agents (RMB) across all goal spaces (Welch’s t-tests at 40k iterations, $p < 0.04$). Agents following a hard-coded curriculum (FC) reached higher median performances than AMB agents on every goal spaces.

Fig. 8.13 shows exploration results in the robotic environment. Agents self-organizing their curriculum (Active Model Babbling) based on their learning progress explore better than agents choosing random objects (Random Model Babbling) in the joysticks, ball, light and sound spaces (Welch’s t-tests at 100k iterations, $p < 0.05$), and better than agents with a hard-coded curriculum sequence (FC) in the right joystick and sound spaces. Agents always choosing to explore the ball (SGS) and agents without a modular representation of goals (FRGB) have low exploration results in all spaces.

Overall, the goal sampling based on learning progress (AMB) improves exploration of most objects of each environment compared to a random choice of object (RMB), as those agents focus on objects that are learnable, ignore the distractor objects and reduce the relative interest of objects already explored for some time. Specifying the curriculum by hand results in a very bad exploration if the agent always directly focuses on an object hard to discover, however if we carefully design the learning sequence given our knowledge of the task, then the final exploration results are similar to autonomous AMB agents.

8.3.4 Influence of the Modularity of Exploration Mutations

The efficiency of the Stepping-Stone Preserving Mutation operator (`SSPMutation`, see Sec. 8.2.2) relies on its ability to preserve the part of the movement that reaches a stepping-stone in the environment, and explore only after the target object started to move in the previous movement being mutated. For instance, the movement would grab the tool without modification, and explore once the controlled toy started to move. To illustrate this mechanism, let us look at actual mutations depending on the mutation operator. Fig. 8.14 shows one movement that reached the magnet tool together with one mutation of this movement, for each mutation operator. The red trajectories are the traces of the gripper (with a circle when open and a point when closed), and the blue trajectories are the traces of the magnet stick. We also plot the initial position of the arm and the magnet stick. We see that in the case of the `SSPMutation` operator, the two red trajectories start to diverge only when the stick is grasped such that the mutated movement also grasps the stick, whereas with the `FullMutation`, the mutation starts right from the beginning of the movement, which in this case makes the mutated movement miss the stick.

We measure how many times the agents succeed to move a tool when they are exploring it, or to move a toy when they are exploring the toy, depending on the mutation operator. Fig. 8.15 shows the proportion of the iterations that allowed to

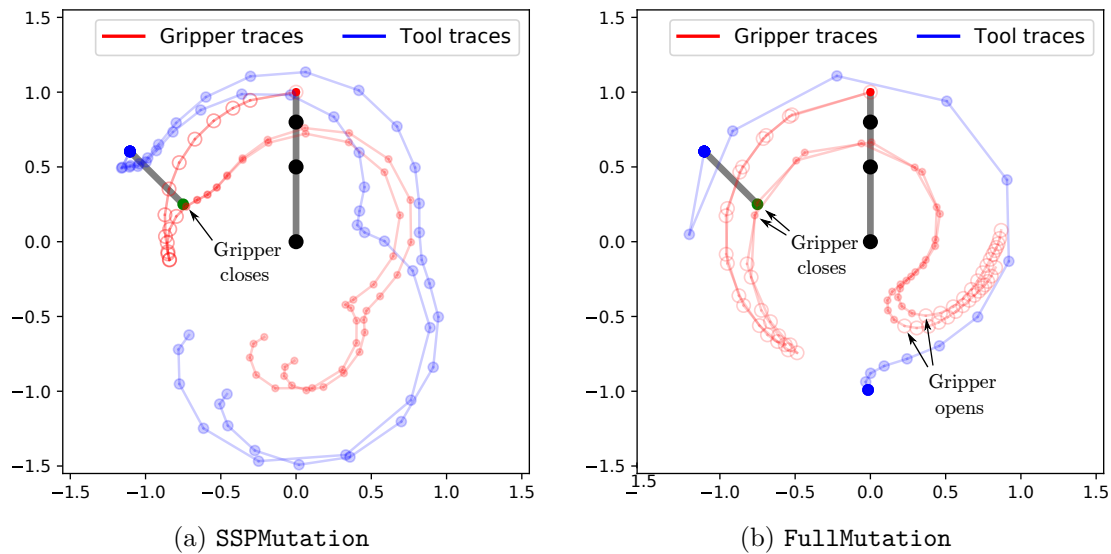


Figure 8.14: Example of a mutation of a movement reaching the tool, for each mutation operator. With the `SSPMutation` operator, the two gripper trajectories start to diverge only when the stick is grasped such that the mutated movement also grasps the stick, whereas with the `FullMutation`, the mutation starts right from the beginning of the movement, which in this case makes the mutated movement miss the stick.

(a) move the magnet tool when this tool is the goal object, (b) move the magnet toy when this toy is the target object, with 50 different runs in the 2D simulated environment (individual runs and median). We can see that with the `FullMutation` operator, at the end of the runs agents succeed to move the tool in 7% of iterations targeted at exploring this tool, versus 95% for the `SSPMutation` operator, and to move the toy in 0.9% of iterations targeted at exploring this toy versus 53%.

The ability of `SSPMutation` to explore while still moving the target object with a high probability directly improves exploration. Fig. 8.16 shows the exploration results of AMB agents with the `SSPMutation` or `FullMutation` operators in the 2D simulated environment in 100 runs with different seeds. The exploration results of the `FullMutation` operator are much lower for the magnet tool (median 13% vs 62%) and magnet toy (median 0% vs 3%, max 0.5% vs 60%).

8.3.5 Encoding of Goals

After executing a movement, the agent receives a sensory feedback containing information about the movement of objects in the environment. The agent then uses the sensory space as an encoding for sampling new goals to reach. In the 2D simulated

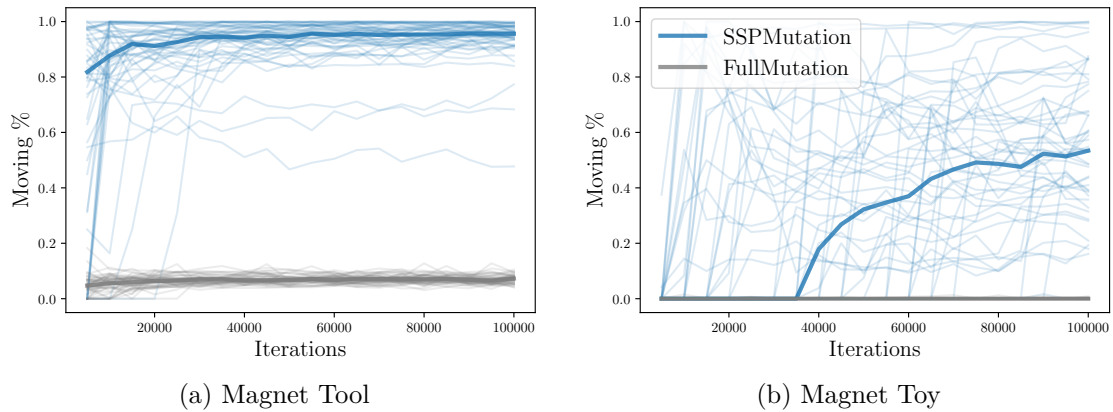


Figure 8.15: Comparison of the **SSPMutation** and **FullMutation** mutation operators. We show the proportion of iterations that allowed to (a) move the magnet tool while exploring this tool, and (b) move the magnet toy while exploring this toy, with 50 different seeds and median. With the **FullMutation** operator, at the end of the runs agents succeed to move the tool in 7% of iterations versus 95% for the **SSPMutation** operator, and to move the toy in 0.9% of iterations versus 53%.

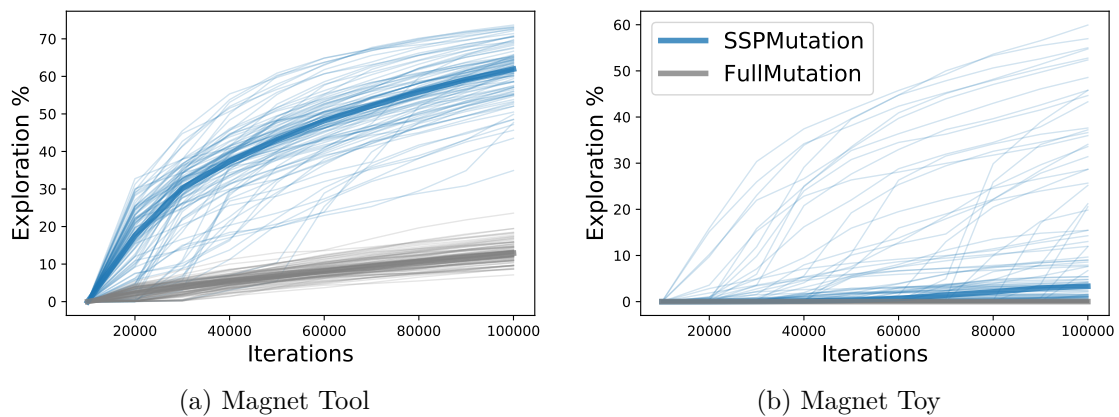


Figure 8.16: Exploration efficiency depending on the mutation operator. **FullMutation** results in a much lower exploration for the magnet tool and toy compared to the stepping-stone preserving operator.

environment, we defined the sensory feedback as the position of each object at the end of the movement. In the robotic environment, as the joysticks may move during the movement but come back by themselves at their rest position, we used a sensory feedback containing information about the whole trajectory of objects as the sequence of their position at 5 time steps during the movement. In this section, we study the influence of the goal encoding on exploration efficiency in the 2D simulated environment.

Fig. 8.17 shows the exploration evolution of AMB agents depending on the encoding of goals: with object trajectories or end points, in the 2D simulated environment. The exploration is a measure of the proportion of cells reached in a discretization of the space of the last position of each object. The encoding with the end position of each object resulted in a slightly better exploration than with trajectories for the magnet tool, and a similar median for the magnet toy but with more variance: with standard deviation of 17% vs 8.6% at 100k iterations for the magnet toy. The trajectory encoding represents the whole trajectory of each object instead of the final point only. This is not strictly needed to represent if a tool or a toy has moved in this environment so the end point encoding may be more efficient once the objects are discovered, however the trajectory encoding helps to explore trajectories with more diversity, for the hand or other objects, and thus to discover hard objects in the first hand. With the trajectory encoding, the exploration of objects difficult to move the first time is thus slower once discovered, but they are more often discovered.

Fig. 8.18 shows examples of interest curves with the goal encoding using trajectories of objects. As the goal spaces are of much larger dimensionality using the object trajectories than with the end position, it takes a longer time to cover the whole sensory space with reached trajectories such that the self-computed interest to explore the hand is higher than with end positions (comparing with Fig. 8.9(top)) and the interest in all spaces takes more time to decrease.

The trajectory encoding is more interesting in environments where the full trajectory of a tool is of importance to control an object, such as in our robotic environment where joysticks come back at their rest position by themselves such that their end position is not informative to predict the end position of the controlled object. We thus use this trajectory encoding in the robotic environment, but we use the end point encoding in the Minecraft Mountain Cart environment.

8.3.6 Static and Random Distractors

In the three tool-use environments, we included distractor objects to harden exploration as those objects can't be controlled by the agent but are however part of their sensory feedback: some of them are static, and some of them move independently of the agent. The Active Model Babbling agents monitor their learning progress to move objects, such that their estimation of their progress to move static object is zero, and

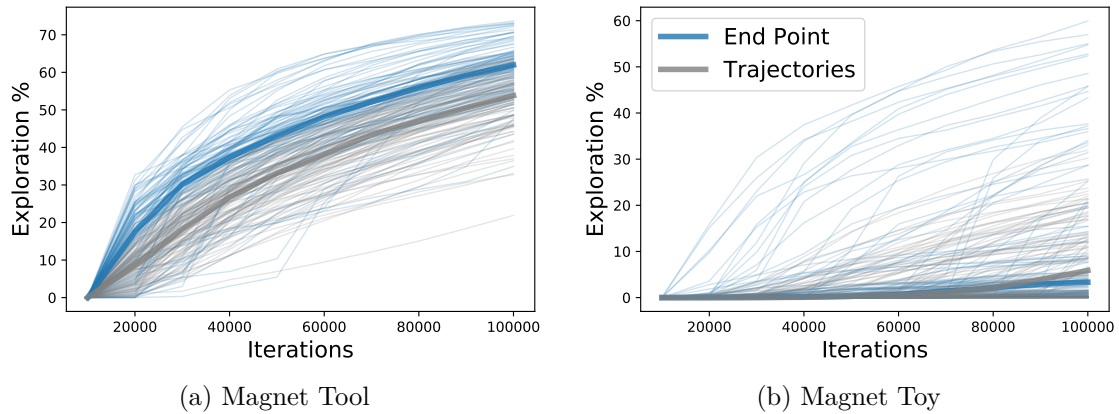


Figure 8.17: Exploration in 2D simulated environment depending on the goal encoding. The encoding with the end position of objects resulted in a slightly better exploration than with trajectories for the magnet tool, and a similar median for the magnet toy but with more variance: standard deviation of 17% vs 8.6% at 100k iterations for the magnet toy.

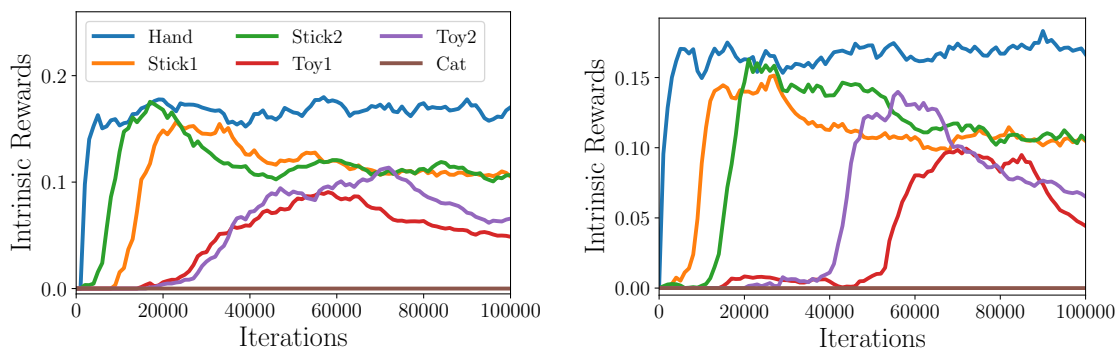


Figure 8.18: Examples of intrinsic rewards in the 2D simulated environment, with an encoding of goals as object trajectories. The sensory spaces are higher-dimensional and take more iterations to be well covered such that the learning progress decreases.

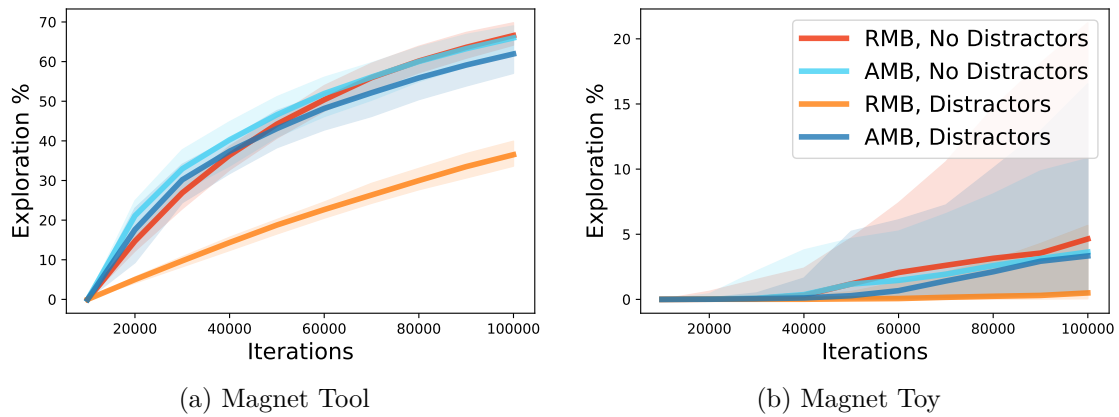


Figure 8.19: Exploration efficiency of Active Model Babbling and Random Model Babbling depending on the presence of distractors in the 2D Simulated environment. The median of 100 runs is plotted together with a shaded area representing the 25-75 percentiles. The efficiency of RMB agents decreases when we add distractors, whereas AMB agents, through their self-estimation of their learning progress to move each object, focus on learnable objects despite having distractors in the environment.

to move other uncontrollable objects is low compared to controllable objects. Here we evaluate the exploration efficiency of AMB and RMB agents in the 2D simulated environment in the presence and absence of distractors to evaluate their robustness to distractors.

Fig 8.19 shows the exploration results depending on the learning condition (RMB vs AMB) and environment condition: with 2 random distractors and 8 static distractors (Distractors) vs without distractors (No Distractors), in the magnet tool and toy spaces (median and 25/75 percentiles of 100 seeds). The RMB agents show a similar exploration without distractors compared to AMB agents. However, we can see that the RMB agents do not cope with distractors while AMB agents do not show a significant decrease in exploration when distractors are added. The learning progress monitoring is thus an efficient mean to discriminate learnable objects from other objects and thus to focus exploration by choosing most goals for learnable objects.

8.4 Discussion

In this chapter, we evaluated different implementations of Intrinsically Motivated Goal Exploration Processes in three tool-use environments: a 2D simulation, a robotic setup and a Minecraft environment. We designed the first real robotic experiment where an intrinsically-motivated humanoid robot discovers a complex continuous high-dimensional environment and succeeds to explore and learn from scratch that some

objects can be used as tools to act on other objects. We also created the Minecraft Mountain Cart environment, where the first intrinsically motivated Minecraft agent learned to use a pickaxe to mine blocks. The IMGEP architecture automatically generated a learning curriculum in the three experimental setups, including a real humanoid robot that can explore multiple spaces of goals with several hundred continuous dimensions. While no particular target goal is provided to the system, this curriculum allows the discovery of skills that act as stepping stones for learning more complex skills, e.g. nested tool use.

Our results show that the algorithms that make use of the modularity of the sensory space based on the objects in the environment explore much better than the agents that use a global representation of all objects (Flat). Furthermore, when the agent monitors its learning progress with intrinsic rewards (AMB), it autonomously develops a learning sequence, or curriculum, from the easiest to the most complex tasks. With those intrinsic rewards (AMB), agents explore more efficiently than without (RMB) when the environment contains objects with different interests for learning, such as some controllable objects and some distractors. Also, the comparison between agents only exploring one problem space set by the user (SGS) versus all spaces (AMB) shows that if an engineer were to specify the target problems to solve (e.g. move the ball), then it would be more efficient to also explore all other possible intrinsic goals to develop new skills (control the joysticks) that can serve as stepping stones to solve the target problems. Agents with intrinsic rewards were developing an autonomous learning curriculum, with a focus on the simplest objects at the beginning, shifting towards the more complex when the progress is high enough, while ignoring the non-controllable objects. In the 2D simulation and the real robotic setup, this algorithm (AMB) was as good as the condition where an expert curriculum was hard-coded (FC) from the simplest to the most complex objects and ignoring the non-controllable ones. It was almost as good as FC in the Minecraft environment.

In chapter 6, the exploratory mutations were performed by the policy Π_ϵ by adding a Gaussian noise on the parameters θ of a low-level motor policy, which had the effect of modifying the full motor trajectory. However, when exploring movements that use a tool to control a toy, modifying the part of the movement where the tool is grasped results in many chances to miss the tool. In this chapter, we studied the Stepping-Stone Preserving Mutation operator, that mutates the movement only from the moment when the target object starts to move. This led to much better exploration performances in the 2D simulation so we used this mechanism in the robotic environment and in Minecraft Mountain Cart. The results in this chapter thus complement and extend the previous experiments presented in chapter 6, with more diverse tool-use setups including a real robotic setup, more efficient exploratory mutations. However, we did not study here alternative inverse models, we used the simple Nearest Neighbor lookup, but we could implement more sophisticated regressions such as LWLR as we showed they could be more efficient once enough data is gathered. In order to investigate the diversity of the exploration databases,

we used them to reach extrinsic goals in the Minecraft environment. We left this study in the other environments for future work.

Like in chapter 6, our Active Model Babbling implementation chooses the object to explore based on the learning progress to control the objects, and the particular goal for that object is chosen randomly. A more sophisticated choice of goal based on learning progress could be implemented as in the SAGG-RIAC algorithm of Baranes and Oudeyer (2010a). We could also evaluate intrinsic rewards based on a combination of novelty and learning progress instead of learning progress alone, as when a novel object is moved, it can take some time before any progress is made while taking into account its novelty could help bootstrap its learning. It would also be interesting to compare the IMGEP approach to Novelty Search and Quality Diversity approaches in those environments, to understand the particular functioning and benefits of each in driving exploration of tool-use stepping stones.

A current limitation of our setup is that we suppose agents already have a perceptual system allowing them to see and track objects, as well as spaces of representations to encode their transformations. In a recent work we study in simulation the learning of a representation of objects from pixels and its use as a goal space for an intrinsically motivated agent (see Appendix B).

Compared to other approaches, our IMGEP implementation is sample-efficient, with a number of iterations of 20k in the real robotic setup, 40k in the Minecraft environment, and 100k in the 2D simulated one. Approaches such as Quality Diversity have been run for 40M iterations for the learning of a hexapod’s locomotion (Cully et al., 2015), and deep Reinforcement Learning agents have required 2M steps in the Atari game Montezuma’s Revenge (Kulkarni et al., 2016), or 50M in a Doom-like first-person shooter (Dosovitskiy and Koltun, 2016). The IMGEP implementation in its simplest form with a Nearest Neighbor look-up for inverse models is also computationally efficient, as we have run the 20k iterations of the real robotic experiment on a raspberry Pi 3.

Chapter 9

Perspectives

9.1 Impact of Intrinsic Motivations on the Interpretation of Child Experiments

In order to evaluate the development of particular sensorimotor skills in infancy, many experimental designs in psychology and neuroscience place objects in the environment and assume this will trigger the corresponding behaviors from the infant. The setup is made attractive to the infant such that he stays concentrated on the task, and other potentially distracting objects are removed from the scene. Behaviors of interest are assessed or neural processes are recorded, which help to understand the mechanisms at play in this task and their changes across development. For instance, in studies evaluating the development of tool-use skills, a colorful toy is placed together with tools in front of the infant, and his attempts and successes to retrieve the toy through the use of tools are measured (see e.g. Chen et al. (2000)). The interpretation of such observations implicitly assumes that the infant wants to achieve the goal of the task and behaves accordingly, displaying his best strategies if the skills are already acquired or doing his best to approach the goal otherwise. The own motivation of infants is therefore mostly neglected in those experiments and their interpretation.

However, infants are curious creatures, intrinsically motivated to explore and learn what they find interesting. They might already have acquired a particular sensorimotor skill and yet not behave to solve the corresponding task. They could display the expected behavior in training trials but not in testing trials or vice versa if it suits them. If they don't know how to solve the task in a particular context but are able to discover it, they may explore and solve the task, but they could as well explore other parts of the setup or the environment. In the case where the infant is obviously not interested in the task, the trial may be excluded from the analysis by the experimenter for "fussiness" or "lack of motivation". Otherwise, the trials are included in the analysis, while the infant may be exploring the experimental setup with goals in mind that are different from the target expected goal, or with the same goal but without the desire to solve it as fast or as best as possible. Depending on the motivation of infants, the particular tasks, measures, and interpretation drawn from the results, the accuracy of the study could suffer in both cases.

The “fussiness” category can hide different behaviors, such as a non-cooperative child, a child not interested in participating, or not interested in the experimental setup but interested in other things in the room. For instance, in visual paradigms with 12 month olds and younger, the rate of missing data for “fussiness” has been 13.7% in studies between 1985 and 2005 (Slaughter and Suddendorf, 2007), out of 22.2% of missing data. As argued by Nicholson et al. (2017), an accurate description of the reasons for exclusion of some data must be provided together with an argument for whether this data is missing completely at random (MCAR), missing at random (MAR) or missing not at random (MNAR). The data is MCAR if the missingness is unrelated to the observed and unobserved explanatory variables. If some known or observed variables can be controlled in the analysis resulting in the missingness being unrelated to the explanatory variables, the data is said to be MAR. If the relation between the causes of missing data and the explanatory variables is unknown, the data is MNAR. In principle, a missingness that depends on the explanatory variables can bias the results of the analysis. This happens for instance if the attrition rate in a longitudinal study depends on the variables, e.g. when the recovery rate during a treatment for a disease is regularly evaluated on the same subjects but some subjects quit the study because they recover before the end of the study. Mechanisms to handle the missing data in each case are reviewed in Nicholson et al. (2017), but a prerequisite is a proper reporting of the protocol for exclusion of data and arguments on its correlation with the measures. However, in developmental studies, the definition of “fussiness” or a protocol for exclusion is almost never provided (Slaughter and Suddendorf, 2007), such that the missingness of the data is implicitly assumed not to bias the results. As argued in chapter 3, infants can have their own goals, different from the task goal set by the experimenter. The “fussiness” category could therefore include trials where infants seem not interested in the task but are actually interested in other aspects of the setup or the environment, which could be because they already master the task or are too bad, or for other unknown reasons. The missingness of the data can be correlated with typical measures of success in the task, as infants with different intrinsic motivations in their objects or intensity, would explore differently the experimental setup and would have a different propensity to be rated as “fussy”. To be able to assess the impact of intrinsic motivations on the missing data and a potential bias in the results of experiments in psychology and neuroscience, a first step would be to explicitly provide the protocol for excluding data and considering the possibility of a relation between data missingness and the variables of interest.

When the data is included in the analysis, the intrinsic motivations of infants to explore the experimental setup can also impact the interpretation of the experimental results. In chapter 3, we reanalyzed an experiment run by Lauriane Rat-Fischer et al, where 21-month olds have to retrieve a toy stuck inside a tube, by inserting several blocks in sequence into the tube. In order to understand the mechanisms of the motivations of babies, we studied in detail their behaviors, goals and strategies in this experiment. We showed that their motivations are diverse and do not always

coincide with the target goal expected and made salient by the experimenter. For instance, many infants seemed to pursue the goal of inserting all blocks into the tube independently of the goal of retrieving the toy, which had the unexpected effect of pushing the toy out of the tube and solving the task.

We also gave in chapter 3 another example of study where intrinsic motivation could interfere with the success results (Koslowski and Bruner, 1972). In this experimental setup, the toy is attached to the side of a lever that is not reachable at the beginning of the experiment, but can be made reachable by rotating the lever. They tested children from 12 to 24-month old, and observed several behaviors: children trying to get the toy directly with the hand, oscillating or partially rotating the lever, playing with the rotation while ignoring the toy even if it got within reach, or rotating the lever and grasping the toy. This lever task has similarities with our tube task in its structure. Indeed, when the child does not directly solve the task, the exploration of one accessible part of the apparatus (the accessible side of the lever) can make the toy come closer, at which point the infant can directly grasp the toy. The reported behaviors are similar to infants in the tube task inserting all objects into the tube, which as a side effect can bring the toy within reach, in which case some of them grasp and play with the toy, while others ignore it or place the toy back into the tube continuing their action. The authors consider only one “goal”, the one decided by the experimenter: getting the toy, or the toy itself, while other behaviors that ignore that goal are “preoccupations”.

In both this lever task and our tube task, the measured success rates can be driven by several factors including the tool-use skills, the interest in getting the toy, the diversity of exploration. Intrinsically motivated exploration plays an important role in the observed behaviors and can interfere with the success rates and their interpretation. Considering the intrinsic motivations of infants and their potential alternative goals and strategies at the time of the design of the experimental setup could help to define measures of success adapted to the research question and therefore facilitate the interpretation of the results.

9.2 Experimental Paradigms for Studying Intrinsic Motivations in Children

Understanding the details of the mechanisms of intrinsic motivations and their relation to extrinsic motivations is an interdisciplinary challenge. Gathering behavioral and neuronal data across the development of diverse skills is a key to make progress on this question. In particular, the mechanisms of the choice of goals and strategies by children are largely unknown.

In chapter 3, we showed that infants explore a diversity of goals and strategies which often exceeds the imagination of the experimenter and the limits of experimental design. To be able to catch and study the intrinsic motivations of infants, the

experimental setup should cover a large proportion of the curiosity space of infants so that they express their curiosity inside the setup and its scope of measures. The setup should therefore provide, when doable, a diversity in objects, such as colors and shapes, in the types of contingencies, in the sensory modalities (visual, auditory, tactile), in the levels of difficulties.

For the study of tool use, the apparatus could provide a number of degrees of freedom related to the use of tools: several salient toys, different tool-use solutions to potential goals, and different possible difficulties in the same setup. For instance, the apparatus could include a box with several toys somehow locked inside, with several different ways to unlock and retrieve those toys, e.g. similar to the Multi Access Box Paradigm of Auersperg et al. (2011) allowing multiple tool-use solutions to the problem of retrieving food in bird studies.

Intrinsic motivations can also push infants to interact with their caregiver, for instance to ask for help or to get feedback. In chapter 5, we modeled a natural interactive play scenario with a caregiver to study the development of vocalizations. In many experiments evaluating particular sensorimotor skills, despite the caregiver being asked not to interact with the infant to avoid perturbing or helping the infant, there is still a significant number of maternal interference leading to missing data (e.g. 0.8% in visual paradigms, Slaughter and Suddendorf (2007)). This is not necessarily at the caregiver's initiative but can be the infant looking for interaction. Interacting with the caregiver could be considered as an exploration strategy that may provide help or feedback on the current behaviors of the infant. The study of intrinsic motivations could take into account the caregiver or an experimenter interacting with the infant as part of the design, allowing the investigation of associated goals and strategies. This could be helpful not only to understand the development of the linguistic function or of social interaction but also the development of any other sensorimotor skill such as tool use.

Investigating intrinsic motivations needs observing in details the behavior of the infant and all the changes in the environment. Indeed, we have shown in chapter 3 that analyzing the behaviors of infants with a fine-grained time scale and taking into account all their actions, their gaze, and the related events in the environment is useful to understand the interaction between intrinsic motivations and the task progress in a tool-use experiment. Also, to facilitate encoding the behavior of the infant and the changes in the environment, digital recording techniques could be used such as head-mounted gaze tracking (Cognolato et al., 2018), pose estimation (Cao et al., 2018), and other sensors recording the movements of the different objects in the scene.

9.3 Tool Use and Language: an Evolutionary Developmental Perspective

Tool use and language seem to require similar information processing capabilities allowing the production and perception of sequential combinations of increasing complexity, from reaching to spoon self-feeding and from vocalizing phonemes to stories. Greenfield (1991) describes how both tool use and language display a hierarchical organization, and draws a parallel between the early development of the tool-use skills and the phonetic skills in the first two years of life that shows the same increases in complexity around the same age. In particular, Meltzoff (1988) relates the development of means-end behaviors and success/failure words and show that their onset is close in time (13 days on average). An insightful use of tools was a better predictor of the use of success/failure words than to predict other abilities such as object-permanence skills. Language and communicative gestures have also been seen as social tools (Borghi et al., 2013; Cohen and Billard, 2018). Experimental evidence shows that the reaching space of a subject can be extended after the use of words, in a setup where an object could be reached with several strategies including the use of a word which triggers an action from another person (Borghi et al., 2013). Deaf children use language gestures as tools, for instance to get others to do things for them (Goldin-Meadow, 2007), when hearing children would have used sentences.

Tool use and language might also share neural correlates. A first link between hand gestures and speech production supports the idea of related neural substrates between speech and hand gestures. Gentilucci et al. (2001) shows that when human subjects are asked to grasp an object and open the mouth, the lip aperture and aperture velocity are higher when the grasped object is large than when it is small. If the subjects have to pronounce a syllable, this is also influenced by a parallel grasping movement. In the case where grasping movements are not executed but observed, they have also been shown to influence speech production Gentilucci (2003). These behaviors are thought to involve the mirror neuron system (Rizzolatti and Craighero, 2004) where neural cells have been shown to both respond if an action is observed in others and respond when that action is executed by the subject. Higuchi et al. (2009) studied brain activations during language and tool tasks in human adults with functional MRI. They found an overlap of activity in both tasks in the dorsal part of area BA44 in Broca's area. This region has previously been reported to be used in complex hierarchical sequential processing in language, such as embedded sentences, but not for sentences without hierarchical structure (Sakai, 2005). According to those results, complex hierarchical structures present both in tool use and language, could be processed in part by the same neural circuits.

How the human species evolved those complex tool-use and language abilities is one of the great mysteries of science. Many evolutionary scenarios have been speculated based on multidisciplinary evidence from archaeology, paleoanthropology,

neuroscience, genetics, ethology, cognitive sciences and computer modeling. The scenarios describe possible paths of evolution of the genes, brains, behaviors and ecology of the hominid lineage. One hypothesis found in many scenarios is that a selection pressure for complex tool use, language and social behaviors have together driven the increase in neural capabilities (Greenfield, 1991; Higuchi et al., 2009; Morgan et al., 2015).

Two and a half millions years ago, stone age's hominins were producing sharp flakes through striking a cobble core with a hammerstone (Morgan et al., 2015), and those sharp flakes were then used as cutting tools, e.g. for butchering. This Oldowan technology was geographically spread and continuously used with little changes for 700,000 years, before the advent of the Acheulean technology including more complex and diverse hand-axe tools. Morgan et al. (2015) argues from experiments with modern humans that in the Oldowan period, tool making and use could have been transmitted through imitation, while teaching and a proto-language could have been prerequisites for the transmission of the Acheulean technology. Bickerton (1990) assigns the Oldowan technology to *homo habilis*, while the *homo ergaster* started the Acheulean manufacture in Africa before its spread in Asia and Europe with *homo erectus* and its descendants. He also hypothesizes that a proto-language was used by *homo erectus* and improved with its evolution, before the advent of language with *homo sapiens*. A proto-language would use a symbolic lexicon but almost no syntax, with sentences composed of a juxtaposition of subjects, nouns and verbs with no particular order, allowing the transmission of simple factual information.

Higuchi et al. (2009) argue that the ability for processing hierarchically organized behaviors was present for tool use in our common ancestors with primates and was later exapted to support language in humans. Greenfield (1991) proposes that the common ancestor of humans and today's primates had the neural circuitry in the left frontal lobe to support both primitive object combinations and primitive language functions, and that they evolved together in the human lineage. Better tool-use abilities would have increased the adaptive value of proto-linguistic communications, and vice versa, both would have evolved through mutually reinforced natural selection. The adaptiveness of language and tool use would have driven the expansion of the prefrontal cortex in a co-evolutionary loop.

Iriki and Taoka (2012) propose that language and tool-use cognitive abilities evolved from the computational processes involved in the control of reaching actions. The authors describe the interdependencies between the ecological, neural and cognitive niches for the human lineage, together called a *triadic niche*. Reaching actions, in particular in the context of bipedalism, imposed a high demand on multi-sensory integration and complex coordinate transformation, that selected brains with improved neural circuitry for processing them. In turn, those neural capabilities could have been reused for other cognitive processes such as the processing of simple tool use and proto-language, which improved the evolutionary fitness of hominins. The development of tool use and language modified the ecological niche which then selected for

more efficient neural circuits. The co-evolution of the ecological, neural and cognitive niches could have slowly enabled and improved higher cognitive functions like tool use and language.

However, in all those evolutionary scenarios, the specification of several key processes is missing: the bootstrapping of evolutionary directions, the transitions between phases of evolution, and the evolutionary pressure inside evolutionary phases. In a simple example of natural selection, when particular swimming patterns or skin features can improve the speed of a fish and as a result its ability to feed and reproduce, the adaptive value of speed imposes an evolutionary pressure that may lead to the emergence of those features on the long term through the selection of random gene mutations. On the other hand, if complex interactions exist between genes, culture, neural and cognitive processes and their ontogenetic development, then the emergence of the adaptive behaviors is not an obvious long-term consequence of the evolutionary pressure. If reaching, gestures or bipedalism select hominids with better neural processes that could be reused for simple tool-use and proto-language, why would a first hominid spontaneously use tools or a proto-language in a culture where those are missing? In a hominid culture with Oldowan tool use, why would brains endowed by chance with better learning processes required for Acheulean tool use would actually improve tool use if nobody is here to demonstrate or teach those improvements, even if those would be selected by the evolutionary pressure? If by chance a particular culture and ecology provides a rich environment for a given period, with a diversity of stone users and uses allowing the discovery and learning of improved tool use, but no particular genetic mutations happen in that period, wouldn't the environment go back to average before selecting improved neural processes?

If the current genome allows the development of complex tool-use and language but the current cultural environment is not adequate, those skills may not develop optimally. In the extreme case where a modern human is deprived from linguistic stimulation for many years, as with the child Genie (Fromkin et al., 1974), the development of language skills can be severely impaired. Genie was deprived from any linguistic input by its parents until the age of 13 when she was discovered and subsequently raised in a foster family. She was then able to acquire a large vocabulary and to string words in simple syntactic constructions but she never fully acquired english grammar (Curtiss, 1977). On the other hand, even if a brain allows the neural and cognitive processes for a particular skill, such a potential skill may not be in use and culturally transmitted, and thus may not drive evolutionary selection. Indeed, experiments with captive chimpanzees showed that they can learn a proto-language when taught by humans (Gardner and Gardner, 1969). Washoe produced 350 signed words of the American Sign Language, was able to string them together to form sentences and to teach some signs to her adopted son Loulis. Chimpanzees thus have the cognitive skills for learning and teaching a proto-language to some extent, and are otherwise great tool makers (Van Lawick-Goodall, 1971). However their lineage did not evolve towards the cultural transmission in the wild of a tool technology and

of proto-language.

Intrinsic motivations could have played a crucial role in the evolution of skills such as tool use and language by pushing organisms to explore, discover and learn the skills that their current body, neural circuitry, cognitive processes, and cultural environment allow. One particularity of the human species and to a lesser extent of other great apes, is the long period of protected development from childhood to adulthood (Power, 1999). The human species may have evolved in the direction of being less capable as a newborn but more capable to learn during development, assisted by an evolution of brain capabilities and an increased role of intrinsic motivations.

Passingham and Wise (2012) studied the evolution of the prefrontal cortex from early primates to anthropoids, and reconstructed the probable ecological niches of the human lineage. Based on its connections with other areas, they argue that the granular prefrontal cortex can represent three hierarchies: the context, goal, and outcome hierarchies. For instance, goals can be represented in a range of hierarchical levels, with goals such as the specification of an object or location used as a target of action, the specification of the abstract structure of a series of actions, or the specification of a rule or strategy that generates objects or locations to choose or to avoid. The prefrontal cortex has the ability to choose actions based on outcomes (medial PFC), choose objects based on outcomes (orbital PFC), search for goals (caudal PFC), generate goals based on recent events (dorsal PFC) and generate goals based on visual and auditory contexts (ventral PFC), such that as a whole, the PFC can generate goals from current contexts and previous events. In the successive ecological niches of primates, the PFC could have been used to link the foraging actions with the resource outcomes that follow, link foraging goals (objects, places) with resource outcomes, select foraging targets, keep goals in memory, allow a fast learning to reduce wasteful and risky choices, and do mental trial-and-error. In the hominid lineage in particular, it could have supported teaching and learning by instruction with less errors, the imagination of more complex goals, the monitoring of others intentions, and improved reasoning abilities.

In chapter 4, we studied how the particular implementations of intrinsic motivations to self-generate interesting goals together with the particular representation of goals can play a role in the tool-use progression of a robotic model. We showed that an intrinsic motivation based on the learning progress to reach goals with a modular representation can self-organize phases of behaviors in the autonomous discovery and development of tool use without pre-existing neural circuits for the processing of means-end actions. In Oudeyer and Smith (2016), the authors explain that previous computational and robotic models of vocalization learning have shown that conventional patterns of vocalizations at the group level could emerge from the interaction of intrinsically motivated individuals. In chapter 5, we presented a robotic model learning both speech and tool use, where the exploration is directed towards self-generated goals in free play, combined with imitation learning of a contingent caregiver. This model does not assume capabilities for complex action sequencing

and combinatorial planning which are often considered necessary for tool use. Yet, we showed that the autonomous exploration of goals allows a learner to progressively discover how to grab objects with the hand, how to use objects as tools to reach further objects, how to autonomously discover new vocal sounds, and how to reproduce the sounds known to other peers and learn their meaning in the environment. A proto-language could thus be culturally transmitted and improved without assuming a pre-existing knowledge of the linguistic function of vocalizations or neural circuitry for the processing of a grammar.

Those models support the idea that intrinsic motivations could have accelerated the evolution of the triadic niche of Iriki and Taoka (2012) in the hominin lineage with a co-evolution of genes and associated neural processes, of cognitive processes, and of cultural environment. Indeed, if individuals with more sophisticated neural capabilities appear from time to time, then the population of hominids intrinsically motivated to explore and learn have more chances to make use of the available capabilities and discover new cognitive skills. With intrinsically motivated exploration, neural processes evolved for reaching or bipedalism could be efficiently exapted for tool use or a proto-language, or capabilities evolved for tool use exapted for language and vice versa. Also, intrinsic motivations can accelerate the evolution of tool use and language thanks to a persistent exploration of tool-use and language skills across generations making use of any available improvement in brain capabilities and in the richness in the cultural environment. Intrinsic motivations could thus help bootstrap evolutionary directions and sustain the impact of the evolutionary pressure.

The evo-devo scientific perspective highlights the complexity of the interaction between the genes and the ontogenetic development of the individual in its environment (Carroll, 2005; Oller et al., 2016). Many genes influence particular neural or cognitive traits by regulating the expression of other genes throughout ontogenetic development. An increasing involvement of intrinsic motivations in the hominid lineage could have been the result of an evolutionary pressure for exploration and learning mechanisms that maximize the use of neural processes throughout ontogenetic development and increase the efficiency of the evolution of brain and cognitive capabilities.

In the case of Genie, intrinsic motivations seem to have quickly reappeared after her introduction to a stimulating and protected environment, and to have facilitated her development of sensorimotor, social and linguistic skills (Fromkin et al., 1974):

Approximately four weeks after her admission to the hospital a consultant described a contrast between her admission status and what he later observed. He wrote that on admission Genie “was pale, thin, ghost-like, apathetic, mute and socially unresponsive. But now she had become alert, bright-eyed, engaged readily in simple social play with balloons, flashlight, and toys, with familiar and unfamiliar adults. She exhibits a lively curiosity, good eye-hand coordination, adequate hearing and vision, and emotional responsiveness. She reveals much stimulus hunger.”

9.4 IMGEP: Follow-ups and Relatives

In chapter 7, we introduced a framework for curiosity-driven exploration through the autonomous generation of goals, called Intrinsically Motivated Goal Exploration Processes (IMGEP). We evaluated particular implementations of this framework in chapter 8, in several experimental setups including a real robotic arm discovering the use of joysticks as tools to move other objects. Exploring diverse spaces of goals with intrinsic motivations was more efficient for learning complex tool-use skills than only trying to directly learn these skills.

A limitation of our setup was that we supposed agents already have a perceptual system allowing them to see and track objects as well as spaces of representations to encode their movements. For instance, the agent in our robotic setup was provided with the trajectory of each object in the environment. In a recent work we studied in simulation the learning of a representation of one object from pixels and its use as a goal space for an intrinsically motivated agent (P  r   et al. (2018), see Appendix B). The agent first learned a representation of the moving object with auto-encoders, and then used this space as a goal space for exploration. Exploration with this learned goal space was as efficient as with a hand-defined goal space adapted to the task (the coordinates of the object). In a more complex environment with several objects, we have shown in chapter 6 that the agent can benefit from a disentangled representation where the variables corresponding to each object can be treated separately in a modular manner. It has been recently shown in Laversanne-Finot et al. (2018) that the agent can learn a disentangled representation and use a learning progress measure to discover the variables corresponding to the different objects. In Reinke et al. (2019), instead of first learning a representation and then using it as a goal space, the representation was learned online, in a settings very different of the learning of inverse models in robotics. With this approach, agents could find self-organized patterns in the complex morphogenetic system *Lenia*, a continuous game-of-life cellular automaton.

In the IMGEP framework, a policy is learned based on the collected data, which given a context and a goal, decides the actions to follow in order to achieve the goal. In population-based IMGEP, based on the context and goal, a high-level policy decides the parameters of a low-level policy to be executed. In our implementation in chapter 8, the low-level policies were defined with open-loop parameterized primitives, and the high-level policy was implemented with a simple nearest neighbor look-up. In chapter 6, we also evaluated a locally-weighted linear regression for the high-level policy, showing slight improvements if enough data was gathered. A limitation of our implementations is that the simple high-level policies, nearest neighbor or regression, are not very accurate, and our open-loop motor primitives do not adapt online to unexpected changes in the environment. Although our results show that this strategy is very sample-efficient for the exploration and discovery of interesting effects in robotic

environments, it does not benefit from the recent advances in Deep Reinforcement Learning useful when a large number of training episodes are available. A recent work has shown that a population-based intrinsically motivated agent within the IMGEP architecture can help bootstrap a deep RL agent (Colas et al., 2018b). Filling a replay buffer with exploratory trajectories collected by an IMGEP algorithm could kick-start the learning of the Deep RL agent by enhancing its exploratory abilities. This approach combines the efficient exploration of population-based IMGEP agents with the efficient fine tuning of policies offered by deep RL agents with a function approximator based on gradient descent.

In our population-based implementations, the high-level policy chooses parameters of low-level parameterized policies, depending on the current goal and context. In Reinforcement Learning, one approach has been to use a population-based policy, such as in Horde (Sutton et al., 2011), where a set of policies is learned each for a different goal. In order to generalize the policy over goals, another approach has been to extend the value function to be parameterized by the goal, such that the policy is learned in a monolithic form, called Universal Value Function Approximators (Schaul et al., 2015). In UVFA, continuous goals are represented through a vector provided as input together with the state to the neural network of the policy and the one of the value function. In Andrychowicz et al. (2017), the agent transfers learning between goals through the use of an experience replay buffer filled with trajectories with different goals than the goal used at exploration time. In Dosovitskiy and Koltun (2016), goals are predefined combinations of measurements (such as ammo, health and frags in the Doom video game). In these lines of work, goals are not modular, and are considered extrinsic to the agent, with extrinsic rewards that can contain expert knowledge about the tasks being learned. Intrinsic rewards have been used in a recent work with an extension of Universal Value Function Approximators with a modular goal representation (Colas et al., 2018a). CURIOS is a particular implementation of the IMGEP architecture using a unique monolithic (multi-task multi-goal) policy network, that learns from a replay buffer filled with rollouts on task and goals of high learning progress. We did not compare our population-based implementation with the CURIOS algorithm as it was developed after.

In the context of sparse rewards, the concept of learning “auxiliary tasks” is also related to intrinsically motivated goal exploration in the sense that tasks that are not directly related to the learning of a final one are trained and help learning. Several auxiliary tasks have been implemented and shown to be useful for learning, such as the control of pixel changes, or the control of intermediate features of a neural network used for policy or value prediction (Jaderberg et al., 2016; Riedmiller et al., 2018).

9.5 IMGEP: Next Steps and Challenges

The IMGEP framework allows the expression of a diversity of learning algorithms that have in common a strong reliance on the exploration of non-optimal behaviors to discover interesting unexpected stepping-stones that can be built upon. This include learning in the presence of sparse rewards, multi-task learning, open-ended or developmental learning. There are a number of challenges for future research on these topics.

A first challenge is to design algorithms that can learn in realistic human-like environments with rich, unpredictable, noisy, continuous, and high-dimensional sensorimotor contingencies. Most learning algorithms have been evaluated in toy environments, with discrete low-dimensional state and action spaces, a limited number of contingencies to explore, a protected resettable environment, and most of the time in simulation. Although the evaluation in such environments is a really useful step in the investigation and understanding of the mechanisms of learning, the agents must also be studied *in the wild*. One aspect of learning which then becomes critical is its scalability in terms of sample-efficiency. Children necessitate relatively few interaction time to learn skills such as tool-use or language in their whole complexity, compared to the millions of iterations or huge databases required by artificial agents to learn a fraction of those skills. The IMGEP implementation of chapter 8 requires a few hours of interaction to learn simple tool-use skills such as controlling a joystick to move a toy in the real robotic setup given an already functioning perceptual system. Some approaches can require millions of iterations in other setups, e.g. 40M iterations for the learning of a hexapod’s locomotion with Quality Diversity (Cully et al., 2015), 2M steps for Deep Reinforcement Learning in the Atari game Montezuma’s Revenge (Kulkarni et al., 2016), or 50M in a Doom-like first-person shooter (Dosovitskiy and Koltun, 2016). To model or get inspiration from the development of infants, one can implement maturational constraints on the body or the processing capabilities (Baranes and Oudeyer, 2011), and investigate assumptions of robotic priors (Jonschkowski and Brock, 2015).

A second challenge is the learning of a representation and generation of goals that can support the efficient exploration of a changing environment whose statistics is shaped by the increasingly complex learned skills. Indeed, in complex environments where advanced skills can be built up incrementally through the recursive combination of stepping-stone skills, the statistics of the gathered sensorimotor data evolves across learning and development, and the explored goals should adapt accordingly. Human children undergo tremendous representational changes in their development of a multitude of skills (Karmiloff-Smith, 1992), where processes of representational redescription are assumed to regularly take place, e.g. in the development of tool use (Guerin et al., 2013). An implementation of online representation learning in the IMGEP framework is given in Reinke et al. (2019), where the representation is period-

ically updated, in an exploration-exploitation alternation cycle. However, designing representational learning architectures that can support an increase in skill complexity and combinations while not forgetting the previously acquired representations and skills (Kirkpatrick et al., 2017) is an important venue for future work.

A third challenge is the integration of intrinsic motivations and social guidance in a sustained positive interaction loop that scaffolds learning. Social guidance can take multiple forms, such as reinforcement, demonstrations, input for imitation, goals to reach, instructions or explanations, and be available from several humans and artificial peers. The combination of intrinsic motivations and reinforcement have been studied in the RL framework. In Riedmiller et al. (2018), an intrinsic motivation for the exploration of auxiliary tasks is combined with a reinforcement at the level of the loss function, such that the agent learns a policy optimizing both at the same time. The combination of intrinsic motivations with demonstrations and imitation of a social peer have for instance been implemented in SGIM-ACTS (Nguyen and Oudeyer, 2012), where the agent can explore autonomously and mimic or emulate demonstrations of several peers. It chooses hierarchically what to learn, how to learn, and from which teacher to get demonstrations based on learning progress heuristics. However, the agent is somehow given prior knowledge about the interactive scenario, as it knows there are movements to be imitated, and teachers that provide information. In chapter 5, we designed an interactive scenario where a caregiver reacts to the learner's actions in two ways: if the robot touches a toy, the caregiver says the name of the toy, and if the robot produces a sound close enough to the name of a toy, the caregiver pushes that toy within reach of the agent. The learner is exploring with two different mechanisms, on one hand with intrinsic motivations to reach many goals consisting in moving objects and producing sounds, and on the other hand imitating the vocal input from the caregiver. It is not given the knowledge that there is a peer or that the sounds produced by the caregiver have a meaning in the environment. The design of a learning algorithm that can start with intrinsic motivations and imitation and little by little learn to benefit from more complex interactive scenarios where a social peer would give targets, instructions or explanations without hard-coding the learning possibilities is an open question.

Appendix A

Tool-Use Experiment: Ethograms

We provide here a table of all coded behaviors, and several additional ethograms.

Subject	Behavior	Modifiers	Description
Experimenter	Task	Trail Tube	Trail task or Tube task (reminder: only one task is coded in one observation, i.e. 2 observations per infant
	Phase	Training Test	Training phase (when available, in each task 1/4 of infants had no training) or Testing phase
	Trial	Trial1 Trial2 Trial3 Trial4 InBetweenTrial	Trial number (from 1 to max 3), or in between trial
	ManipApparatus	Hold Move/Turn PlaceToolBack	Experimenter holds or moves/turns the apparatus to prevent infant from turning or lifting the apparatus too much, or to cheat by accessing the reward directly with its hand after lifting/turning apparatus. Experimenter places the tool back next to or on top of the apparatus after it was discarded or it failed away
	AttractAttention	Talking Pointing Talking&Pointing	Experimenter attracts the attention of the subject towards the toy or solving the task, e.g. "where is the car?", "can you get the car out?", "what can you do?", "HOW can you get the car out", "can you try something else?"
	Encourage		The attention of the subject is on toy and/or tool, the experimenter encourage the infant, praise him for what he is currently doing or what he just did
	Spoiler		Experimenter shows or tell the baby how to solve the problem, e.g. "Push!"
	Surprise		Experimenter shows to the baby that she is surprised
Caregiver	AttractAttention	Talking Pointing Talking&Pointing	Caregiver attracts the attention of the subject towards the toy or solving the task, e.g. "where is the car?", "can you get the car out?", "what can you do?", "HOW can you get the car out", "can you try something else?"
	Encourage		The attention of the subject is on toy and/or tool, the caregiver encourage the infant, praise him for what he is currently doing or what he just did
	Surprise		Caregiver shows to the baby that she is surprised
Baby	Apparatus	Hand Mouth	Manipulate/explore the apparatus with hands/mouth
	Combine	IpsilateralHand ContralateralHand ; Insert Dip Probe Hit Scratch Touch Stack Push/Pull PutUnder ; Right Side Left Side Both Sides ; Apparatus Toy Environment OtherPerson OwnBody OtherTool	Combines the tool with something
	DiscardTool	GiveTo Throw PutOnSide Drops	Discard/get rid off the tool
	Fussiness		Loose patience (can be accompanied with vocalization)
	HoldTool		Holds the tool (while doing other things or not)
	Perspective	MoveAroundApp TurnApp WalkOverApp LiftApp	Changing the perspective according to apparatus, e.g. moving around the apparatus or turning the apparatus
	PlayTool	SeveralIndependently SeveralTogether OneTool	Play with tool (manipulate/explore the tool)
	PushToolInside		Push a tool inside the tube
	SwapTool-A		Takes another tool straight after discarding another (usually happens after a first unsuccessful trial with a tool, the infant then switches to another to another tool - even though the other tool is of the same shape)
	Vocalizations	Autocongratulation VocalizeToAdult FrustrationVocalization Satisfactory vocalization ExplicitAskForHelp Surprise Other	Autocongrats that can be accompanied by applause, vocalize in order to get help from adult, frustration vocalization, other vocalization

Table A.1: Annotated behaviors.

	WalkAway		Walk away from the apparatus or push (tries to push) the apparatus away
	LookApparatus		If not interacting with tool or toy (but can interact with apparatus), or we don't know exactly what he looks at but in the direction of apparatus
	LookHand		Look at hand while doing nothing else
	LookSocial		Look at the experimenter or the caregiver
	LookTool		Look at a tool, also count when baby manipulating a tool within apparatus
	LookToy		Look at a toy, also count when toy inside apparatus
	DiscardToy	GiveTo Throw PutOnSide Drops ; Experimenter Caregiver Sibling/Other	In between trials: baby discards the toy
	ManipToy	PutInMouth PlayHands	In between trials: baby manipulates the toy
	PlaceToyBack		In between trials: baby places the object back in the apparatus, as if he wanted to start the game again (but it could be for another reason, we don't need to specify))
	ToolToyCombination	Hit Scratch Touch Push/pull Stack Other	In between trials: baby uses the toy and the tool in combination after retrieval
Environment	ToyState		Map: 7 subdivisions in the tube: 1 and 7 are reachable with hand; 0 is out of the tube
Observer	ToolUse-Goal		Infant's goal may be to use the wooden block as a tool
	InsertAll		Infant's goal may be to insert all shapes in holes (infant's goal is to insert anything in the apparatus, usually the objects around that are meant to be the tools; he may then connect the tool and the toy by chance, and switch to the goal "retrieveToy")
	MakeNoise		Infant's goal may be to make noise with the apparatus /tools / toys
	Retrieve	Toy Tool PossiblyTool&Toy	Infant's goal may be to retrieve the toy or the tool outside the apparatus
	Draw-Goal		Infant seems to try to draw with a stick
	G-PlaceToyBack		Place the toy back inside the tube
	Direct	Point Reach	Infant's strategy may be to use its hand to try to reach/retrieve the toy (point, reach)
	Social		Infant's strategy may be to look/ask someone in the room. Infant tries to engage another agent in the process of getting the toy: vocalizes/asks while trying to reach for tool, looks at agent, looks & vocalizes/asks, takes agent's hand
	SwapTool		Infant's strategy may be to try another tool after having tried a first tool, usually unsuccessfully
	ToolUse-Strategy		Infant's strategy may be to use a tool, for different goals: e.g. to reach/retrieve toy, or to make noise
	SwitchHand		Infant switch hand as part of a strategy (using a tool or not)
	SocialTool		Infant takes caregiver's hand and direct it toward the apparatus/toy
	Result	Failure SuccessByChance SuccessIntentional SuccessAfterDemo	According to the observer, outcome of the trial: Failure, (to get toy out), SuccessAfterDemo (only during training, if any), SuccessByChance (also includes non intentional success), SuccessIntentional (if not sure whether intentional or by chance, then be conservative and code "by chance")
	ScoreTrial	1-Fail 2-CombineChance 3-TrialError 4-ImmediateSuccess	1= fail with no insertion, 2=insertion, either fail, or succeeds but not intentional, 3=succeeds by trial and error, 4=immediate success)

Table A.1 (Continued): Annotated behaviors.

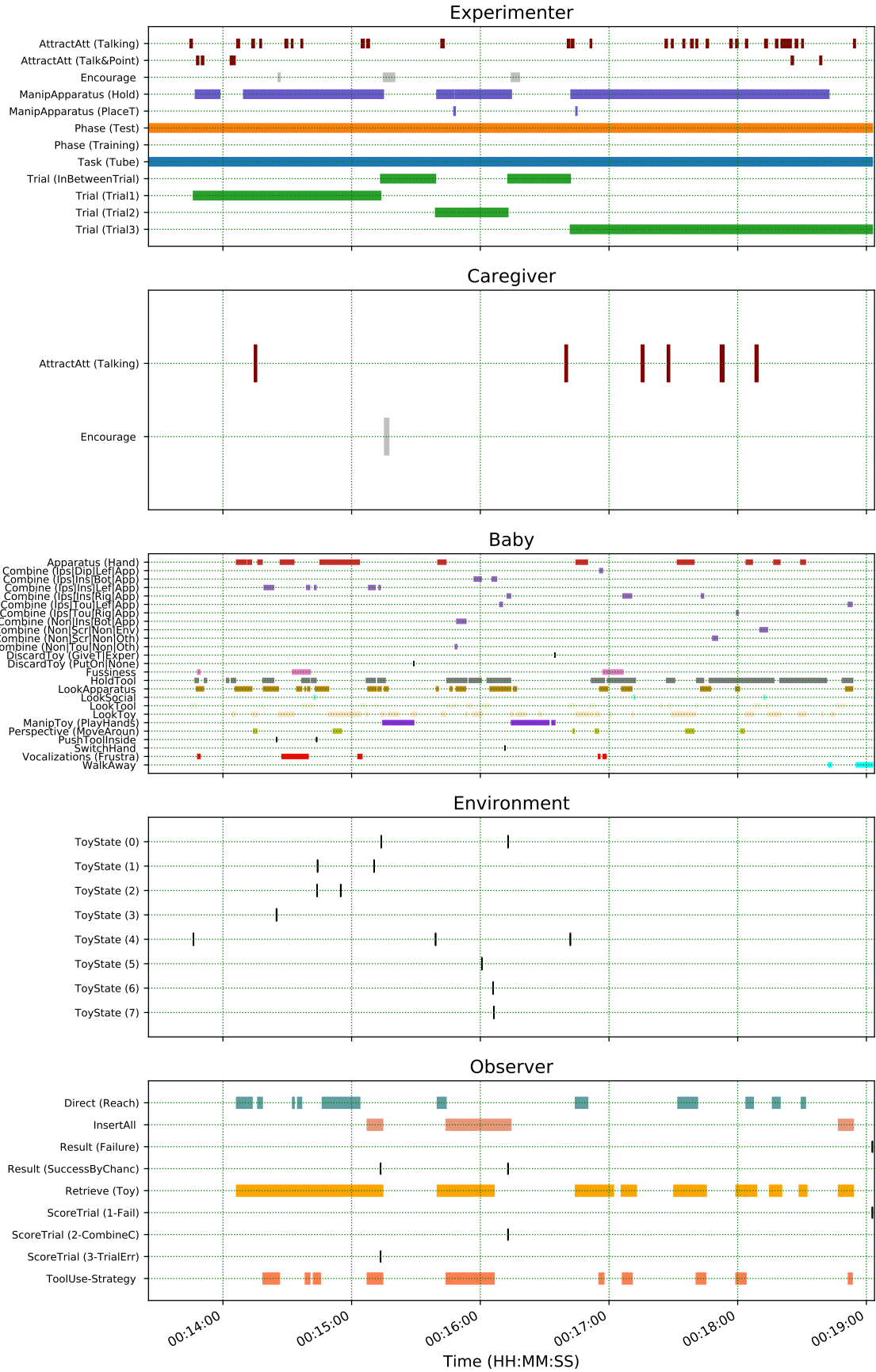


Figure A.1: Ethogram of experiment with baby A1.

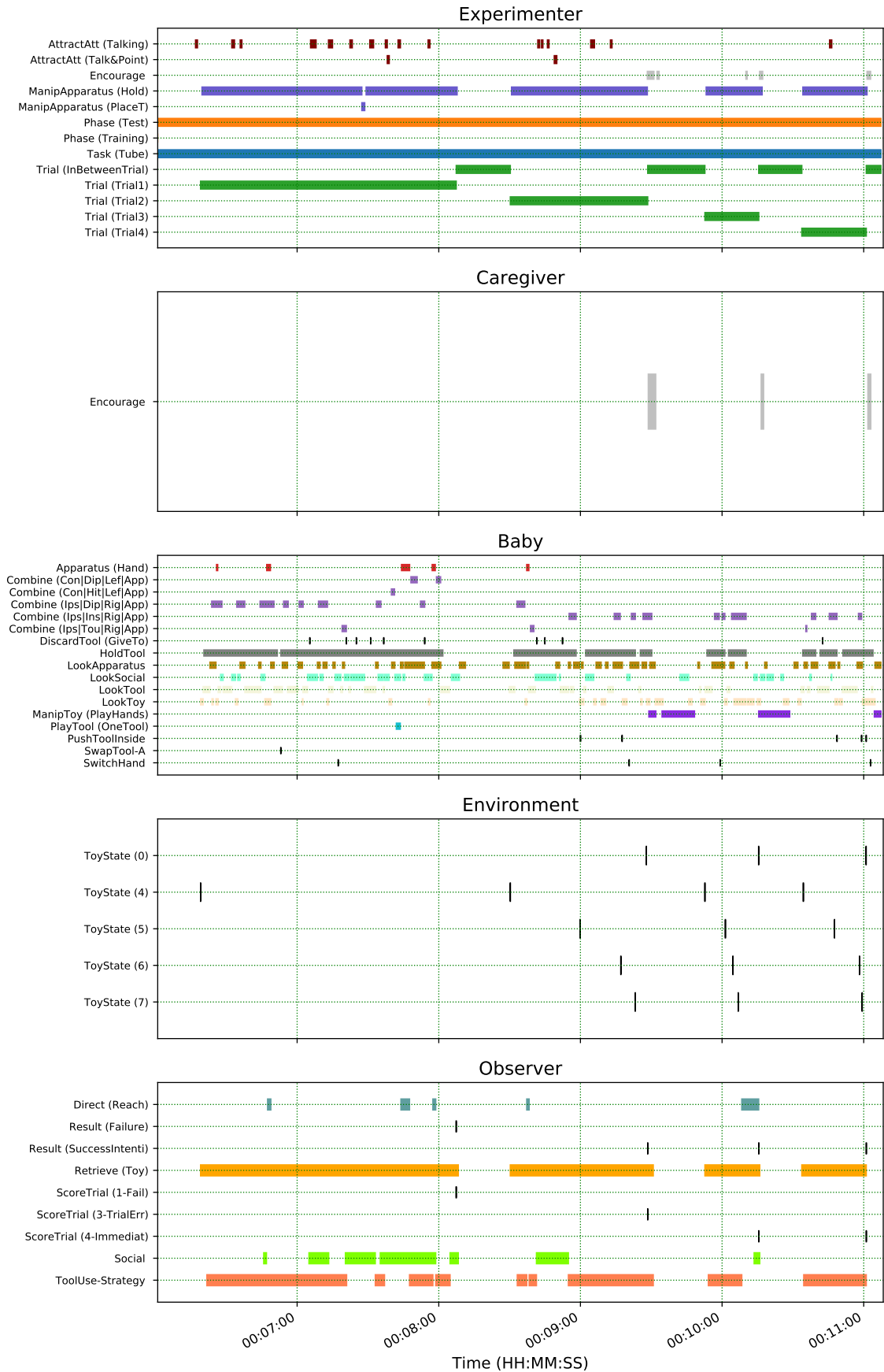


Figure A.2: Ethogram of experiment with baby C2.

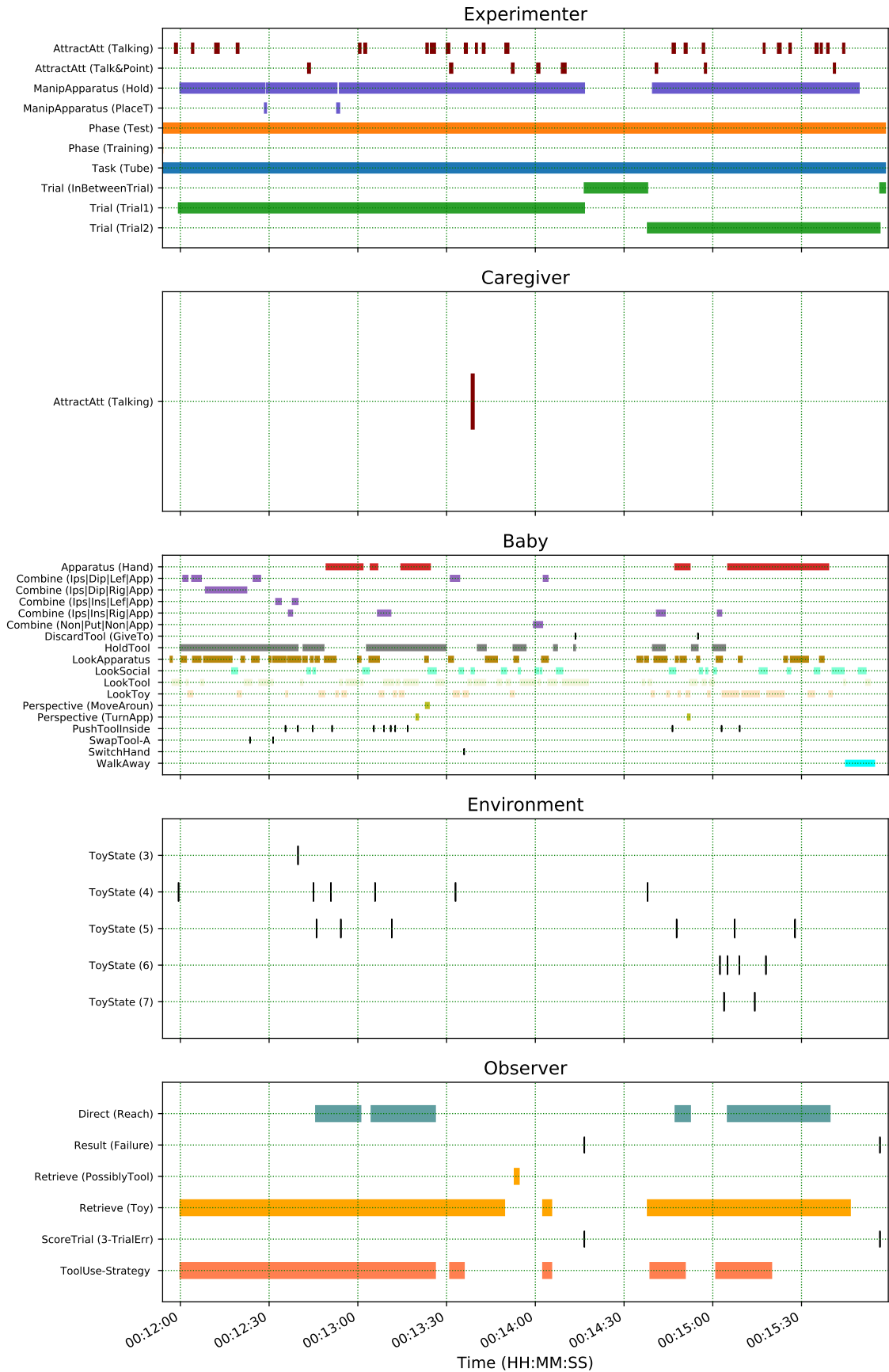


Figure A.3: Ethogram of experiment with baby H1.

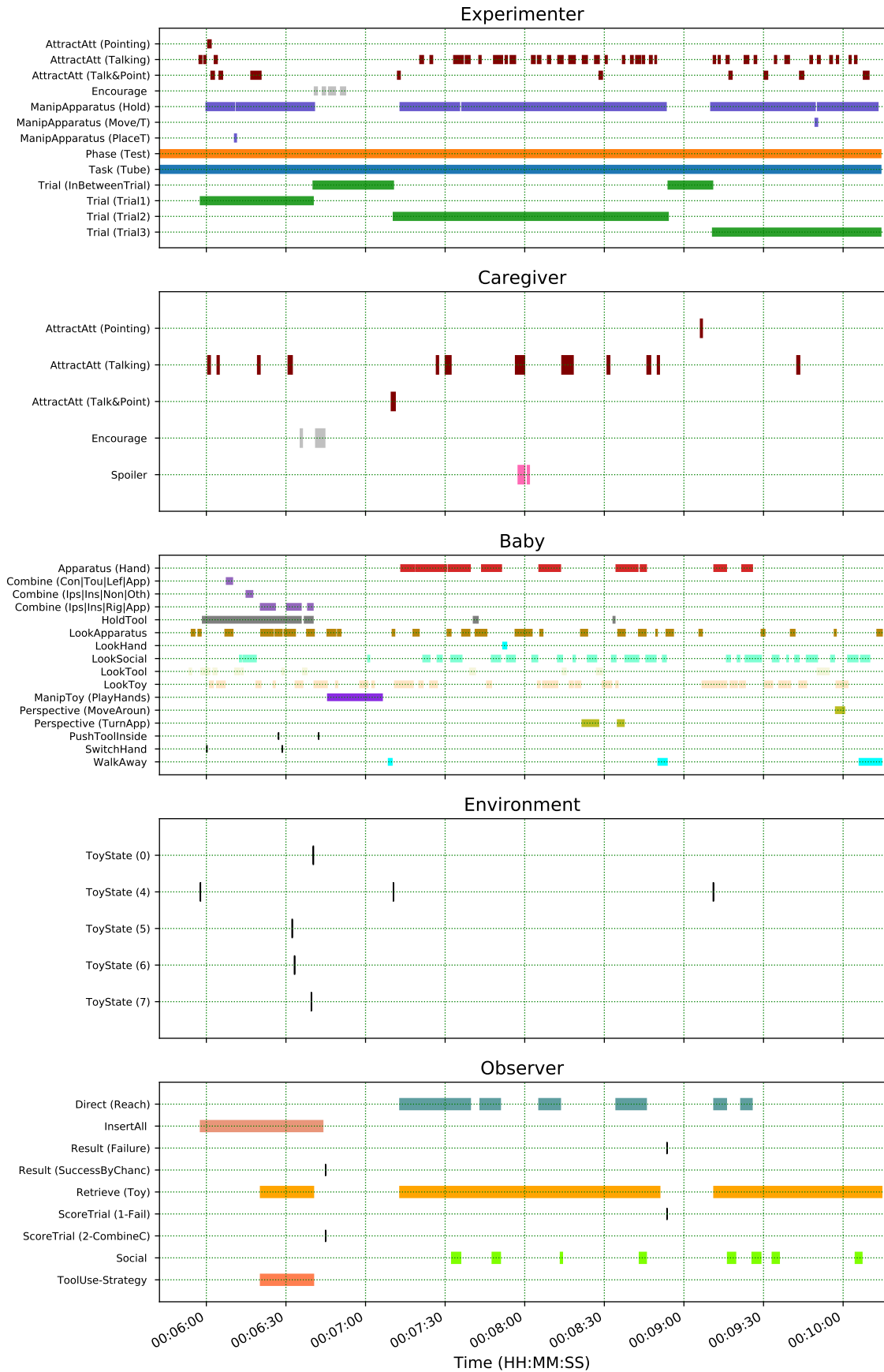


Figure A.4: Ethogram of experiment with baby L3.

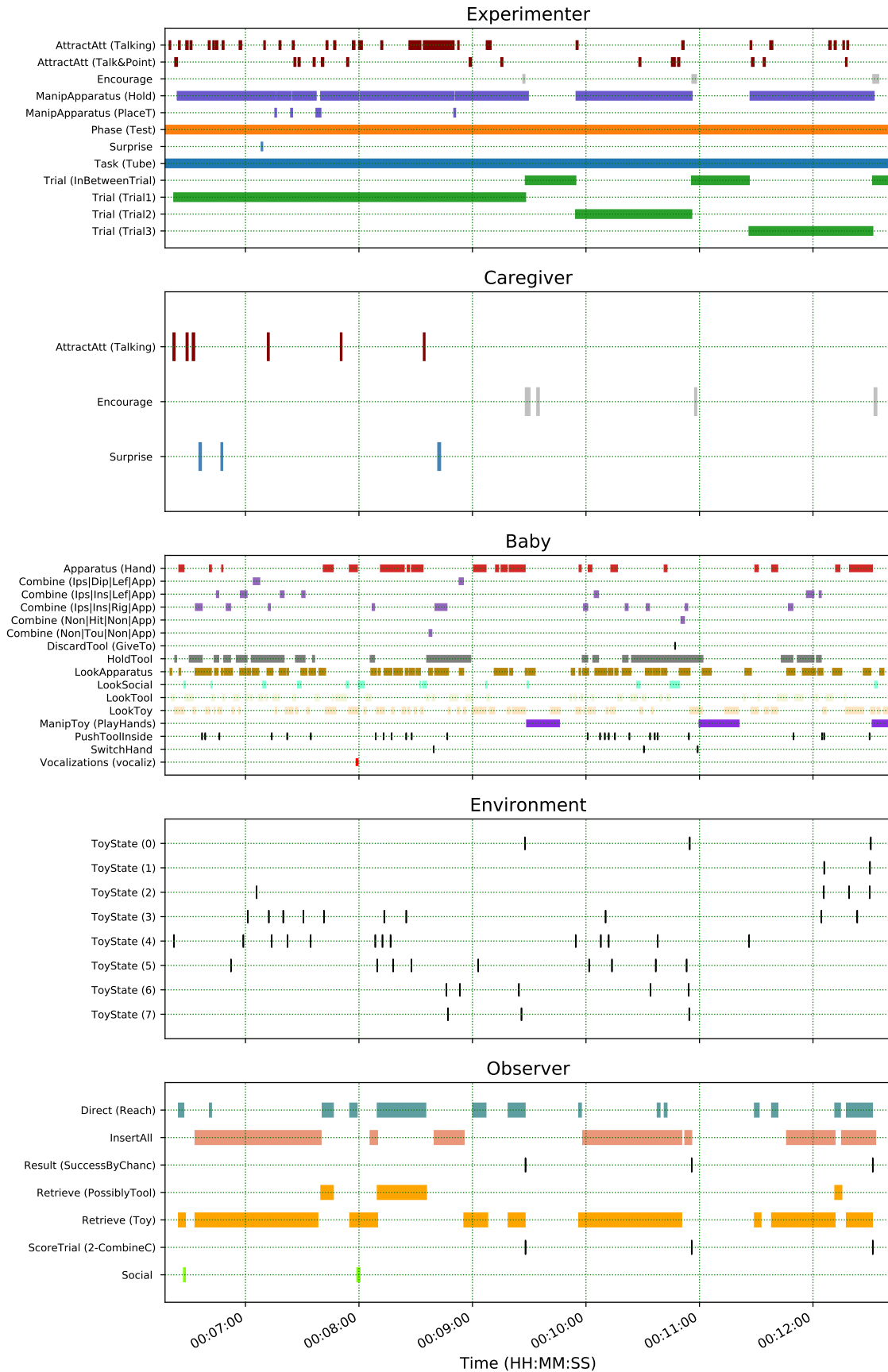


Figure A.5: Ethogram of experiment with baby O1.

Appendix B

Learning a Representation for Goal Babbling

We reproduce here the main text of Péré et al. (2018) for the convenience of the reader. This work has been done and written mainly by Alexandre Péré, during an internship co-supervised by Pierre-Yves Oudeyer and myself.

Unsupervised Learning of Goal Spaces for Intrinsically Motivated Goal Exploration

Alexandre Péré Sébastien Forestier Olivier Sigaud
Pierre-Yves Oudeyer

Abstract

Intrinsically motivated goal exploration algorithms enable machines to discover repertoires of policies that produce a diversity of effects in complex environments. These exploration algorithms have been shown to allow real world robots to acquire skills such as tool use in high-dimensional continuous state and action spaces. However, they have so far assumed that self-generated goals are sampled in a specifically engineered feature space, limiting their autonomy. In this work, we propose to use deep representation learning algorithms to learn an adequate goal space. This is a developmental 2-stage approach: first, in a perceptual learning stage, deep learning algorithms use passive raw sensor observations of world changes to learn a corresponding latent space; then goal exploration happens in a second stage by sampling goals in this latent space. We present experiments where a simulated robot arm interacts with an object, and we show that exploration algorithms using such learned representations can match the performance obtained using engineered representations.

Keywords: exploration; autonomous goal setting; diversity; unsupervised learning; deep neural network

1 Introduction

Spontaneous exploration plays a key role in the development of knowledge and skills in human children. For example, young children spend a large amount of time exploring what they can do with their body and external objects, independently of external objectives such as finding food or following instructions from adults. Such intrinsically motivated exploration (Berlyne, 1966; Gopnik et al., 1999; Oudeyer & Smith, 2016) leads them to make ratcheting discoveries, such as learning to locomote or climb in various styles and on various surfaces, or learning to stack and use objects as tools. Equipping machines with similar intrinsically motivated exploration capabilities should also be an essential dimension for lifelong open-ended learning and artificial intelligence.

In the last two decades, several families of computational models have both contributed to a better understanding of such exploration processes in infants, and how to apply them efficiently for autonomous lifelong machine learning (Oudeyer et al., 2016). One general approach taken by several research groups (Baldassarre et al., 2013; Oudeyer et al., 2007; Barto, 2013; Friston et al., 2017) has been to model the child as intrinsically motivated to make sense of the world, exploring like a scientist that imagines, selects and runs experiments to gain knowledge and control over the world. These models have focused in particular on three kinds of mechanisms argued to be essential and complementary to enable machines and animals to efficiently explore and discover skill repertoires in the real world (Oudeyer et al., 2013; Cangelosi et al., 2015): embodiment¹, intrinsic motivation² and social guidance³. This article focuses on challenges related to learning goal representations for intrinsically motivated exploration, but also leverages models of embodiment, through the use of parameterized Dynamic Movement Primitives controllers (Ijspeert et al., 2013) and social guidance, through the use of observations of another agent.

Given an embodiment, intrinsically motivated exploration⁴ consists in automatically and spontaneously conducting experiments with the body to discover both the world dynamics and how it can be controlled through actions. Computational models have framed intrinsic motivation as a family of mechanisms that self-organize agents exploration curriculum, in particular through generating and selecting experiments that maximize measures such as novelty (Andreae & Andreae, 1978; Sutton, 1990), predictive information gain (Little & Sommer, 2013), learning progress (Schmidhuber, 1991; Kaplan & Oudeyer, 2003), compression progress (Schmidhuber, 2013), competence progress (Baranes & Oudeyer, 2013), predictive information (Martius et al., 2013) or empowerment (Salge et al., 2014). When used in the Reinforcement Learning (RL) framework (e.g. (Sutton, 1990; Schmidhuber, 1991; Kaplan & Oudeyer, 2003; Barto, 2013)), these measures have been called intrinsic rewards, and they are often applied to reward the "interestingness" of actions or states that are explored. They have been consistently shown to enable artificial agents or robots to make discoveries and solve problems that would have been difficult to learn using a classical optimization or RL approach based only on the target reward (which is often rare or deceptive) (Chentanez et al., 2005; Baranes & Oudeyer, 2013; Stanley & Lehman, 2015). Recently, they have been similarly used to guide exploration in difficult deep RL problems with sparse rewards, e.g. (Bellemare et al., 2016; Houthoofd et al., 2016; Tang et al., 2017; Pathak et al., 2017).

However, many of these computational approaches have considered intrinsically motivated exploration at the level of micro-actions and states (e.g. considering low-level actions and pixel level perception). Yet, children's intrinsically motivated ex-

¹Body synergies provide structure on action and perception

²Self-organizes a curriculum of exploration and learning at multiple levels of abstraction

³Leverages what others already know

⁴Also called curiosity-driven exploration

ploration leverages abstractions of the environments, such as objects and qualitative properties of the way they may move or sound, and explore by setting self-generated goals (Von Hofsten, 2004), ranging from objects to be reached, toy towers to be built, or paper planes to be flown. A computational framework proposed to address this higher-level form of exploration has been Intrinsically Motivated Goal Exploration Processes (IMGEPs) (Baranes & Oudeyer, 2009; Forestier et al., 2017), which is closely related to the idea of goal babbling (Rolf et al., 2010). Within this approach, agents are equipped with a mechanism enabling them to sample a goal in a space of parameterized goals⁵, before they try to reach it by executing an experiment. Each time they sample a goal, they dedicate a certain budget of experiments time to improve the solution to reach this goal, using lower-level optimization or RL methods for example. Most importantly, in the same time, they take advantage of information gathered during this exploration to discover other outcomes and improve solutions to other goals⁶.

This property of cross-goal learning often enables efficient exploration even if goals are sampled randomly (Baranes & Oudeyer, 2013) in goal spaces containing many unachievable goals. Indeed, generating random goals (including unachievable ones) will very often produce goals that are outside the convex hull of already discovered outcomes, which in turn leads to exploration of variants of known corresponding policies, pushing the convex hull further. Thus, this fosters exploration of policies that have a high probability to produce novel outcomes without the need to explicitly measure novelty. This explains why forms of random goal exploration are a form of intrinsically motivated exploration. However, more powerful goal sampling strategies exist. A particular one consists in using meta-learning algorithms to monitor the evolution of competences over the space of goals and to select the next goal to try, according to the expected competence progress resulting from practicing it (Baranes & Oudeyer, 2013). This enables to automate curriculum sequences of goals of progressively increasing complexity, which has been shown to allow high-dimensional real world robots to acquire efficiently repertoires of locomotion skills or soft object manipulation (Baranes & Oudeyer, 2013), or advanced forms of nested tool use (Forestier et al., 2017). Similar ideas have been recently applied in the context of multi-goal deep RL, where architectures closely related to intrinsically motivated goal exploration are used by procedurally generating goals and sampling them randomly (Cabi et al., 2017; Najnin & Banerjee, 2017) or adaptively (Florensa et al., 2017).

Yet, a current limit of existing algorithms within the family of Intrinsically Motivated Goal Exploration Processes is that they have assumed that the designer⁷ provides a representation allowing the autonomous agent to generate goals, together

⁵Here a goal is not necessarily an end state to be reached, but can characterize certain parameterized properties of changes of the world, such as following a parameterized trajectory.

⁶E.g. while learning how to move an object to the right, they may discover how to move it to the left.

⁷Here we consider the human designer that crafts the autonomous agent system.

with formal tools used to measure the achievement of these goals (e.g. cost functions). For example, the designer could provide a representation that enables the agent to imagine goals as potential continuous target trajectories of objects (Forestier et al., 2017), or reach an end-state starting from various initial states defined in Euclidean space (Florensa et al., 2017), or realize one of several discrete relative configurations of objects (Cabi et al., 2017), which are high-level abstractions from the pixels. While this has allowed to show the power of intrinsically motivated goal exploration architectures, designing IMGEPs that sample goals from a learned goal representation remains an open question. There are several difficulties. One concerns the question of how an agent can learn in an unsupervised manner a representation for hypothetical goals that are relevant to their world before knowing whether and how it is possible to achieve them with the agent’s own action system. Another challenge is how to sample "interesting" goals using a learned goal representation, in order to remain in regions of the learned goal parameters that are not too exotic from the underlying physical possibilities of the world. Finally, a third challenge consists in understanding which properties of unsupervised representation learning methods enable an efficient use within an IMGEP architecture so as to lead to efficient discovery of controllable effects in the environment.

In this paper, we present one possible approach named IMGEP-UGL where aspects of these difficulties are addressed within a 2-stage developmental approach, combining deep representation learning and goal exploration processes:

Unsupervised Goal space Learning stage (UGL) In the first phase, we assume the learner can passively observe a distribution of world changes (e.g. different ways in which objects can move), perceived through raw sensors (e.g. camera pixels or other forms of low-level sensors in other modalities). Then, an unsupervised representation learning algorithm is used to learn a lower-dimensional latent space representation (also called embedding) of these world configurations. After training, a Kernel Density Estimator (KDE) is used to estimate the distribution of these observations in the latent space.

Intrinsically Motivated Goal Exploration Process stage (IMGEP) In the second phase, the embedding representation and the corresponding density estimation learned during the first stage are reused in a standard IMGEP. Here, goals are iteratively sampled in the embedding as target outcomes. Each time a goal is sampled, the current knowledge (forward model and meta-policy, see below) enables to guess the parameters of a corresponding policy, used to initialize a time-bounded optimization process to improve the cost of this policy for this goal. Crucially, each time a policy is executed, the observed outcome is not only used to improve knowledge for the currently selected goal, but for all goals in the embedding. This process enables the learner to incrementally discover new policy parameters and their associ-

ated outcomes, and aims at learning a repertoire of policies that produce a maximally diverse set of outcomes.

A potential limit of this approach, as it is implemented and studied in this article, is that representations learned in the first stage are frozen and do not evolve in the second stage. However, we consider here this decomposition for two reasons. First, it corresponds to a well-known developmental progression in infant development: in their first few weeks, motor exploration in infants is very limited (due to multiple factors), while they spend a considerable amount of time observing what is happening in the outside world with their eyes (e.g. observing images of social peers producing varieties of effects on objects). During this phase, a lot of perceptual learning happens, and this is reused later on for motor learning (infant perceptual development often happens ahead of motor development in several important ways). Here, passive perceptual learning from a database of visual effects observed in the world in the first phase can be seen as a model of this stage where infants learn by passively observing what is happening around them⁸. A second reason for this decomposition is methodological: given the complexity of the underlying algorithmic components, analyzing the dynamics of the architecture is facilitated when one decomposes learning in these two phases (representation learning, then exploration).

Main contribution of this article. Prior to this work, and to our knowledge, all existing goal exploration process architectures used a goal space representation that was hand designed by the engineer, limiting the autonomy of the system. Here, the main contribution is to show that representation learning algorithms can discover goal spaces that lead to exploration dynamics close to the one obtained using an engineered goal representation space. The proposed algorithmic architecture is tested in two environments where a simulated robot learns to discover how to move and rotate an object with its arm to various places (the object scene being perceived as a raw pixel map). The objective measure we consider, called KL-coverage, characterizes the diversity of discovered outcomes during exploration by comparing their distribution with the uniform distribution over the space of outcomes that are physically possible (which is unknown to the learner). We even show that the use of particular representation learning algorithms such as VAEs in the IMGEP-UGL architecture can produce exploration dynamics that match the one using engineered representations.

Secondary contributions of this article:

- We show that the IMGEP-UGL architecture can be successfully implemented (in terms of exploration efficiency) using various unsupervised learning algo-

⁸Here, we do not assume that the learner actually knows that these observed world changes are caused by another agent, and we do not assume it can perceive or infer the action program of the other agent. Other works have considered how stronger forms of social guidance, such as imitation learning (Schaal et al., 2003), could accelerate intrinsically motivated goal exploration (Nguyen & Oudeyer, 2014), but they did not consider the challenge of learning goal representations.

gorithms for the goal space learning component: AutoEncoders (AEs) (Bourlard & Kamp, 1988), Variational AE (VAE) (Rezende et al., 2014; Kingma & Ba, 2015), VAE with Normalizing Flow (Rezende & Mohamed, 2015), Isomap (Tenenbaum et al., 2000), PCA (Pearson, 1901), and we quantitatively compare their performances in terms of exploration dynamics of the associated IMGEP-UGL architecture.

- We show that specifying more embedding dimensions than needed to capture the phenomenon manifold does not deteriorate the performance of these unsupervised learning algorithms.
- We show examples of unsupervised learning algorithms (Radial Flow VAEs) which produce less efficient exploration dynamics than other algorithms in our experiments, and suggest hypotheses to explain this difference.

2 Goals Representation learning for Exploration Algorithms

In this section, we first present an outline of intrinsically motivated goal exploration algorithmic architectures (IMGEPs) as originally developed and used in the field of developmental robotics, and where goal spaces are typically hand crafted. Then, we present a new version of this architecture (IMGEP-UGL) that includes a first phase of passive perceptual learning where goal spaces are learned using a combination of representation learning and density estimation. Finally, we outline a list of representation learning algorithms that can be used in this first phase, as done in the experimental section.

2.1 Intrinsically Motivated Goal Exploration Algorithms

Intrinsically Motivated Goal Exploration Processes (IMGEPs), are powerful algorithmic architectures which were initially introduced in Baranes & Oudeyer (2009) and formalized in Forestier et al. (2017). They can be used as heuristics to drive the exploration of high-dimensional continuous action spaces so as to learn forward and inverse control models in difficult robotic problems. To clearly understand the essence of IMGEPs, we must envision the robotic agent as an experimenter seeking information about an unknown physical phenomenon through sequential experiments. In this perspective, the main elements of an exploration process are:

- A *context* c , element of a Context Space \mathcal{C} . This context represents the initial experimental factors that are not under the robotic agent control. In most cases, the context is considered fully observable (e.g. state of the world as measured by sensors).
- A *parameterization* θ , element of a Parameterization Space Θ . This parameterization represents the experimental factors that can be controlled by the

robotic agent (e.g. parameters of a policy).

- An *outcome* o , element of an Outcome Space \mathcal{O} . The outcome contains information qualifying properties of the phenomenon during the execution of the experiment (e.g. measures characterizing the trajectory of raw sensor observations during the experiment).
- A *phenomenon dynamics* $D : \mathcal{C}, \Theta \mapsto \mathcal{O}$, which in most interesting cases is unknown.

If we take the example of the *Arm-Ball* problem⁹ in which a multi-joint robotic arm can interact with a ball, the context could be the initial state of the robot and the ball, the parameterization could be the parameters of a policy that generate a sequence of motor torque commands for N time steps, and the outcome could be the position of the ball at the last time step. Developmental roboticists are interested in developing autonomous agents that learn two models, the forward model $\tilde{D} : \mathcal{C} \times \Theta \mapsto \mathcal{O}$ which approximates the phenomenon dynamics, and the inverse model $\tilde{I} : \mathcal{C} \times \mathcal{O} \mapsto \Theta$ which allows to produce desired outcomes under given context by properly setting the parameterization. Using the aforementioned elements, one could imagine a simple strategy that would allow the agent to gather tuples $\{c, \theta, o\}$ to train those models, by uniformly sampling a random parameterization $\theta \sim \mathcal{U}(\theta)$ and executing the experiment. We refer to this strategy as *Random Parameterization Exploration*. The problem for most interesting applications in robotics, is that only a small subspace of Θ is likely to produce interesting outcomes. Indeed, considering again the Arm-Ball problem with time-bounded action sequences as parameterizations, very few of those will lead the arm to touch the object and move it. In this case, a random sampling in Θ would be a terrible strategy to yield interesting samples allowing to learn useful forward and inverse models for moving the ball.

To overcome this difficulty, one must come up with a better approach to sample parameterizations that lead to informative samples. Intrinsicly Motivated Goal Exploration Strategies propose a way to address this issue by giving the agent a set of tools to handle this situation:

- A *Goal Space* \mathcal{T} whose elements τ represent parameterized goals that can be targeted by the autonomous agent. In the context of this article, and of the IMGEP-UGL architecture, we consider the simple but important case where the *Goal Space* is equated with the *Outcome space*. Thus, goals are simply vectors in the outcome space that describe target properties of the phenomenon that the learner tries to achieve through actions.
- A *Goal Policy* $\gamma(\tau)$, which is a probability distribution over the Goal Space used for sampling goals (see Algorithmic Architecture 2). It can be stationary, but in most cases, it will be updated over time following an intrinsic motivation strategy. Note that in some cases, this Goal Policy can be conditioned on the context $\gamma(\tau|c)$.

⁹See Section 3 for details.

- A set of *Goal-parameterized Cost Functions* $C_\tau : \mathcal{O} \mapsto \mathbb{R}$ defined over all \mathcal{O} , which maps every outcome with a real number representing the goodness-of-fit of the outcome o regarding the goal τ . As these cost functions are defined over \mathcal{O} , this enables to compute the cost of a policy for a given goal even if the goal is imagined after the policy roll-out. Thus, as IMGEPs typically memorize the population of all executed policies and their outcomes, this enables reuse of experimentations across multiple goals.
- A *Meta-Policy* $\Pi : \mathcal{T}, \mathcal{C} \mapsto \Theta$ which is a mechanism to approximately solve the minimization problem $\Pi(\tau, c) = \arg \min_\theta C_\tau(\tilde{D}(\theta, c))$, where \tilde{D} is a running forward model (approximating D), trained on-line during exploration.

In some applications, a *de-facto* ensemble of such tools can be used. For example, in the case where \mathcal{O} is an Euclidean space, we can allow the agent to set goals in the Outcome Space $\mathcal{T} = \mathcal{O}$, in which case for every goal τ we can consider a Goal-parameterized cost function $C_\tau(o) = \|\tau - o\|$ where $\|\cdot\|$ is a similarity metric. In the case of the Arm-Ball problem, the final position of the ball can be used as Outcome Space, hence the Euclidean distance between the goal position and the final ball position at the end of the episode can be used as Goal-parameterized cost function (but one could equally choose the full trajectories of the ball as outcomes and goals, and an associated similarity metric).

Algorithmic architecture 2 describes the main steps of Intrinsically Motivated Goal Exploration Processes using these tools¹⁰:

Bootstrapping phase: Sampling a few policy parameters (Random Parametrization Exploration, RPE), observing the starting context and the resulting outcome, to initialize a memory of experiments ($\mathcal{H} = \{(c_i, \theta_i, o_i)\}$) and a regressor $\tilde{D}_{running}$ approximating the phenomenon dynamics.

Goal exploration phase: Stochastically mixing random policy exploration with goal exploration. In goal exploration, one first observes the context c and then samples a goal τ using goal policy γ (this goal policy can be a random stationary distribution, as in experiments below, or a contextual multi-armed bandit maximizing information gain or competence progress, see (Baranes & Oudeyer, 2013)). Then, a meta-policy algorithm Π is used to search the parameterization θ minimizing the Goal-parameterized cost function C_τ , i.e. it computes $\theta = \arg \min_\theta C_\tau(\tilde{D}_{running}(\theta, c))$. This process is typically initialized by searching the parameter θ_{init} in \mathcal{H} such that the corresponding c_{init} is in the neighborhood of c and $C_\tau(o_{init})$ is minimized. Then, this initial guess is improved using an optimization algorithm (e.g. L-BFGS) over the regressor $\tilde{D}_{running}$. The resulting policy θ is executed, and the outcome o is observed. The observation (c, θ, o) is then used to update \mathcal{H} and $\tilde{D}_{running}$.

¹⁰IMGEPs characterize an architecture and not an algorithm as several of the steps of this architecture can be implemented in multiple ways, for e.g. depending on which regression or meta-policy algorithms are implemented

This procedure has been experimentally shown to enable sample efficient exploration in high-dimensional continuous action robotic setups, enabling in turn to learn repertoires of skills in complex physical setups with object manipulations using tools (Forestier & Oudeyer, 2016; Forestier et al., 2017) or soft deformable objects (Nguyen & Oudeyer, 2014).

Nevertheless, two issues arise when it comes to using these algorithms in real-life setups, and within a fully autonomous learning approach. First, there are many real world cases where providing an Outcome Space (in which to make observations and sample goals, so this is also the Goal Space) to the agent is difficult, since the designer may not himself understand well the space that the robot is learning about. The approach taken until now (Forestier et al., 2017), was to create an external program which extracted information out of images, such as tracking all objects positions. This information was presented to the agent as a point in $[0, 1]^n$, which was hence considered as an Outcome Space. In such complex environments, the designer may not know what is actually feasible or not for the robot, and the Outcome space may contain many unfeasible goals. This is the reason why advanced mechanisms for sampling goals and discovering which ones are actually feasible have been designed (Baranes & Oudeyer, 2013; Forestier et al., 2017). Second, a system where the engineer designs the representation of an Outcome Space space is limited in its autonomy. A question arising from this is: can we design a mechanism that allows the agent to construct an Outcome Space that leads to efficient exploration by the mean of examples? Representation Learning methods, in particular Deep Learning algorithms, constitute a natural approach to this problem as it has shown outstanding performances in learning representations for images. In the next two sections, we present an update of the IMGEP architecture that includes a goal space representation learning stage, as well as various Deep Representation Learning algorithms tested: Autoencoders along with their more recent Variational counter-parts.

2.2 Unsupervised Goal Representation Learning for IMGEP

In order to enable goal space representation learning within the IMGEP framework, we propose to add a first stage of unsupervised perceptual learning (called UGL) before the goal exploration stage, leading to the new IMGEP-UGL architecture described in Algorithmic Architecture 1. In the passive perceptual learning stage (UGL, lines 2-8), the learner passively observes the unknown phenomenon by collecting samples x_i of raw sensor values as the world changes. The architecture is neutral with regards to how these world changes are produced, but as argued in the introduction, one can see them as coming from actions of other agents in the environment. Then, this database of x_i observations is used to train an unsupervised learning algorithm (e.g. VAE, Isomap) to learn an embedding function $\tilde{\mathcal{R}}$ which maps the high-dimensional raw sensor observations onto a lower-dimensional representation o . Also, a kernel density estimator *KDE* estimates the distribution $p_{kde}(o)$ of observed

world changes projected in the embedding. Then, in the goal exploration stage (lines 9-26), this lower-dimensional representation o is used as the outcome and goal space, and the distribution $p_{kde}(o)$ is used as a stochastic goal policy, within a standard IMGEP process (see above).

2.3 Representation Learning Algorithms and Density Estimation for the UGL stage

As IMGEP-UGL is an algorithmic architecture, it can be implemented with several algorithmic variants depending on which unsupervised learning algorithm is used in the UGL phase. We experimented over different deep and classical Representation Learning algorithms for the UGL phase. We rapidly outline these algorithms here. For a more in-depth introduction to those models, the reader can refer to Appendix B which contains details on the derivations of the different Cost Functions and Architectures of the Deep Neural Networks based models.

Auto-Encoders (AEs) are a particular type of Feed-Forward Neural Networks that were introduced in the early hours of neural networks (Bourlard & Kamp, 1988). They are trained to output a reconstruction $\tilde{\mathbf{x}}$ of the input vector \mathbf{x} of dimension D , through a representation layer of size $d < D$. They can be trained in an unsupervised manner using a large dataset of unlabeled samples $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i \in \{0 \dots N\}}$. Their main interest lies in their ability to model the statistical regularities existing in the data. Indeed, during training, the network learns the regularities allowing to encode most of the information existing in the input in a more compact representation. Put differently, AEs can be seen as learning a non-linear compression for data coming from an unknown distribution. Those models can be trained using different algorithms, the most simple being Stochastic Gradient Descent (SGD), to minimize a loss function $\mathcal{J}(\mathcal{D})$ that penalizes differences between $\tilde{\mathbf{x}}$ and \mathbf{x} for all samples in \mathcal{D} .

Variational Auto-Encoders (VAEs) are a recent alternative to classic AEs (Rezende et al., 2014; Kingma & Ba, 2015), that can be seen as an extension to a stochastic encoding. The argument underlying this model is slightly more involved than the simple approach taken for AEs, and relies on a statistical standpoint presented in Appendix B. In practice, this model simplifies to an architecture very similar to an AE, differing only in the fact that the encoder f_θ outputs the parameters μ and σ of a multivariate Gaussian distribution $\mathcal{N}(\mu, \text{diag}(\sigma^2))$ with diagonal covariance matrix, from which the representation \mathbf{z} is sampled. Moreover, an extra term is added to the Cost Function, to condition the distribution of \mathbf{z} in the representation space. Under the restriction that a factorial Gaussian is used, the neural network can be made fully differentiable thanks to a *reparameterization trick*, making it possible to use SGD for training.

Algorithmic Architecture 1: Intrinsically Motivated Goal Exploration Process with Unsupervised Goal Representation Learning (IMGEP-UGL)

Input:

Regressor $\tilde{D}_{running}$, Goal Policy γ , Parameterized cost function C_τ , Meta-Policy algorithm Π , Unsupervised representation learning algorithm \mathcal{A} (e.g. AE, VAE, Isomap), Kernel Density Estimator algorithm KDE , History \mathcal{H} , Random exploration ratio Γ_e

1 **begin**2 **Passive perceptual learning stage (UGL):**3 **for** *A fixed number of Observation iterations* n_r **do**4 Observe the phenomenon with raw sensors to gather a sample x_i 5 Add this sample to a sample database $\mathcal{D} = \{x_i\}_{i \in [0, n_r]}$ 6 Learn an embedding function $\tilde{R} : x \rightarrow o$ using algorithm \mathcal{A} on data \mathcal{D} 7 Set $\mathcal{O} = \mathcal{T} = \tilde{R}(x)$ 8 Estimate the outcome distribution $p_{kde}(o)$ from $\{\tilde{R}(x_i)\}_{i \in [0, 10000]}$ using algorithm KDE 9 Set the Goal Policy $\gamma = p_{kde}$ to be the estimated outcome distribution10 **Goal exploration stage (IMGEP):**11 **for** *A fixed number of Bootstrapping iterations* **do**12 Observe context c 13 Sample $\theta \sim \mathcal{U}(\theta)$

14 Perform experiment and retrieve outcome from raw sensor signal

 $o = \tilde{R}(x)$ 15 Update Regressor $\tilde{D}_{running}$ with tuple $\{c, \theta, o\}$ 16 $\mathcal{H} = \mathcal{H} \cup \{c, \theta, o\}$ 17 **for** *A fixed number of Exploration iterations* **do**18 **if** $u \sim \mathcal{U}(0, 1) < \Gamma_e$ **then**19 Sample a random parameterization $\theta_i \sim p(\theta)$ 20 **else**21 Observe context c 22 Sample a goal $\tau \sim \gamma$ 23 Compute $\theta = \arg \min_{\theta} C_\tau(\tilde{D}_{running}(\theta, c))$ using Π , $\tilde{D}_{running}$ and \mathcal{H}

24 Perform experiment and retrieve outcome from raw sensor signal

 $o = \tilde{R}(x)$ 25 Update Regressor $\tilde{D}_{running}$ with a tuple $\{c, \theta, o\}$ 26 Update Goal Policy γ , according to Intrinsic Motivation strategy27 $\mathcal{H} = \mathcal{H} \cup \{c, \theta, o\}$ 28 **return** *The forward model $\tilde{D}_{running}$, the history \mathcal{H} and the embedding \tilde{R}*

In practice VAEs tend to yield smooth representations of the data, and are faster to converge than AEs from our experiments. Despite these interesting properties, the derivation of the actual cost function relies mostly on the assumption that the factors can be described by a factorial Gaussian distribution. This hypothesis can be largely erroneous, for example if one of the factors is periodic, multi-modal, or discrete. In practice our experiments showed that even if training could converge for non-Gaussian factors, it tends to be slower and to yield poorly conditioned representations.

Normalizing Flow proposes a way to overcome this restriction on distribution, by allowing more expressive ones (Rezende & Mohamed, 2015). It uses the classic rule of change of variables for random variables, which states that considering a random variable $\mathbf{z}_0 \sim q(\mathbf{z}_0)$, and an invertible transformation $t : \mathbb{R}^d \mapsto \mathbb{R}^d$, if $\mathbf{z} = t(\mathbf{z}_0)$ then $q(\mathbf{z}) = q(\mathbf{z}_0) |\det \partial t / \partial \mathbf{z}_0|^{-1}$. Using this, we can chain multiple transformations t_1, t_2, \dots, t_K to produce a new random variable $\mathbf{z}_K = t_K \circ \dots \circ t_2 \circ t_1(\mathbf{z}_0)$. One particularly interesting transformation is the *Radial Flow*, which allows to radially contract and expand a distribution as can be seen in Figure 5 in Appendix. This transformation seems to give the required flexibility to encode periodic factors.

Isomap is a classical approach of Multi-Dimensional Scaling (Kruskal, 1964) a procedure allowing to embed a set of N -dimensional points in a n dimensional space, with $N > n$, minimizing the *Kruskal Stress*, which measures the distortion induced by the embedding in the pairwise Euclidean distances. This algorithm results in an embedding whose pairwise distances are roughly the same as in the initial space. Isomap (Tenenbaum et al., 2000) goes further by assuming that the data lies in the vicinity of a lower dimensional manifold. Hence, it replaces the pairwise Euclidean distances in the input space by an approximate pairwise geodesic distance, computed by the Dijkstra’s Shortest Path algorithm on a κ nearest-neighbors graph.

Principal Component Analysis is an ubiquitous procedure (Pearson, 1901) which, for a set of data points, allows to find the orthogonal transformation that yields linearly uncorrelated data. This transformation is found by taking the principal axis of the covariance matrix of the data, leading to a representation whose variance is in decreasing order along dimensions. This procedure can be used to reduce dimensionality, by taking only the first n dimensions of the transformed data.

Estimation of sampling distribution: Since the Outcome Space \mathcal{O} was learned by the agent, it had no prior knowledge of $p(o)$ for $o \in \mathcal{O}$. We used a *Gaussian Kernel Density Estimation* (KDE) (Parzen, 1962; Rosenblatt, 1956) to estimate this distribution from the projection of the images observed by the agent, into the learned

goal space representation. Kernel Density Estimation allows to estimate the continuous density function (cdf) $f(o)$ out of a discrete set of samples $\{o_i\}_{i \in \{1, \dots, n\}}$ drawn from distribution $p(o)$. The estimated cdf is computed using the following equation:

$$\hat{f}_{\mathbf{H}}(o) = \frac{1}{n} \sum_{i=1}^n K_{\mathbf{H}}(o - o_i), \quad (1)$$

with $K(\cdot)$ a kernel function and \mathbf{H} a bandwidth $d \times d$ matrix (d the dimension of \mathcal{O}). In our case, we used a Gaussian Kernel:

$$K_{\mathbf{H}}(o) = (2\pi)^{-\frac{d}{2}} |\mathbf{H}|^{-\frac{1}{2}} e^{-\frac{1}{2} o^T \mathbf{H}^{-1} o}, \quad (2)$$

with the bandwidth matrix \mathbf{H} equaling the covariance matrix of the set of points, rescaled by factor $n^{-\frac{1}{d+4}}$, with n the number of samples, as proposed in Scott (1992).

3 Experiments

We conducted experiments to address the following questions in the context of two simulated environments:

- Is it possible for an IMGEP-UGL implementation to produce a Goal Space representation yielding an exploration dynamics as efficient as the dynamics produced by an IMGEP implementation using engineered goal space representations? Here, the dynamics of exploration is measured through the *KL Coverage* defined thereafter.
- What is the impact of the target embedding dimensionality provided to these algorithms?
- Are there differences in exploration dynamics when one uses different unsupervised learning algorithms (Isomap-KDE, PCA-KDE, AE-KDE, VAE-KDE, VAE-GP, RFVAE-GP, RFVAE-KDE) as various UGL component of IMGEP-UGL?

We now present in depth the experimental campaign we performed¹¹.

Environments: We experimented on two different **Simulated Environments** derived from the Arm-Ball benchmark represented in Figure 1, namely the *Arm-Ball* and the *Arm-Arrow* environments, in which a 7-joint arm, controlled by a 21 continuous dimension Dynamic Movement Primitives (DMP) (Ijspeert et al., 2013) controller, evolves in an environment containing an object it can handle and move around in the scene. In the case of IMGEP-UGL learners, the scene is perceived as a 70x70 pixel image. For the UGL phase, we used the following mechanism to generate

¹¹The code to reproduce the experiments is available at https://github.com/flowersteam/Unsupervised_Goal_Space_Learning



Figure 1: Left: The Arm-Ball environment with a 7 DOF arm, controlled by a 21D continuous actions DMP controller, that can stick and move the ball if the arm tip touches it (on the left). Right: rendered 70x70 images used as raw signals representing the end position of the objects for Arm-Ball (on the center) and Arm-Arrow (on the right) environments. The arm is not visible to learners.

the distribution of samples x_i : the object was moved randomly uniformly over $[-1, 1]^2$ for ArmBall, and over $[-1, 1]^2 \times [0, 2\pi]$ for ArmArrow, and the corresponding images were generated and provided as an observable sample to IMGEP-UGL learners. Note that the physically reachable space (i.e. the largest space the arm can move the object to) is the disk centered on 0 and of radius 1: this means that the distribution of object movements observed by the learner is slightly larger than the actual space of moves that learners can produce themselves (and learners have no knowledge of which subspace corresponds to physically feasible outcomes). The environments are presented in depth in Appendix C.

Algorithmic Instantiation of the IMGEP-UGL Architecture: We experimented over the following Representation Learning Algorithms for the UGL component: *Auto-Encoders* with *KDE* (RGE-AE), *Variational Auto-Encoders* with *KDE* (RGE-VAE), *Variational Auto-Encoders* using the associated Gaussian prior for sampling goal instead of *KDE* (RGE-VAE-GP), *Radial Flow Variational Auto-Encoders* with *KDE* (RGE-RFVAE), *Radial Flow Variational Auto-Encoders* using the associated Gaussian prior for sampling goal (RGE-RFVAE-GP), *Isomap* (RGE-Isomap) (Tenenbaum et al., 2000) and *Principal Component Analysis* (RGE-Isomap).

Regarding the classical IMGEP components, we considered the following elements:

- **Context Space $\mathcal{C} = \emptyset$:** In the implemented environments, the initial positions of the arm and the object were reset at each episode¹². Consequently, the context was not observed nor accounted for by the agent.

¹²This makes the experiment faster but does not affect the conclusion of the results.

- **Parameterization Space** $\Theta = [0, 1]^{21}$: During the experiments, we used DMP controllers as parameterized policies to generate time-bounded motor actions sequences. Since the DMP controller was parameterized by 3 basis functions for each joint of the arm (7), the parameterization of the controller was represented by a point in $[0, 1]^{3 \times 7}$.
- **Outcome Space** $\mathcal{O} \subset \mathbb{R}^l$: The Outcome Space is the subspace of \mathbb{R}^l spanned by the embedding representations of the ensemble of images observed in the first phase of learning. For the RGE-EFR algorithm, $l = 2$ in ArmBall and $l = 3$ in ArmArrow. For IMGEP-UGL algorithms, as the representation learning algorithms used in the UGL stage require a parameter specifying the maximum dimensionality of the target embedding, we considered two cases in experiments: 1) $l = 10$, which is 5 times larger than the true manifold dimension for ArmBall, and 3.3 times larger for ArmArrow (the algorithm is not supposed to know this, so testing the performance with larger embedding dimension is key); 2) $l = 2$ for ArmBall, and $l = 3$ for ArmArrow, which is the same dimensionality as the true dimensions of these manifolds.
- **Goal Space** $\mathcal{T} = \mathcal{O}$: The Goal Space was taken to equate the Outcome Space.
- **Goal-Parameterized Cost function** $C_\tau(\cdot) = \|\tau - \cdot\|_2$: Sampling goals in the Outcome Space allows us to use the Euclidean distance as Goal parameterized cost function.

Considering those elements, we used the instantiation of the IMGEP architecture represented in Appendix D in Algorithm 3. We implemented a goal sampling strategy known as *Random Goal Exploration* (RGE), which consists, given a stationary distribution over the Outcome Space $p(o)$, in sampling a random goal $o \sim p(o)$ each time (note that this stationary distribution $p(o)$ is learnt in the UGL stage for IMGEP-UGL implementations). We used a simple k -neighbors regressor to implement the running forward model \tilde{D} , and the Meta-Policy mechanism consisted in returning the nearest achieved outcome in the outcome space, and taking the same parameterization perturbed by an exploration noise (which has proved to be a very strong baseline in IMGEP architectures in previous works (Baranes & Oudeyer, 2013; Forestier & Oudeyer, 2016)).

Exploration Performance Measure: In this article, the central property we are interested in is the dynamics and quality of exploration of the outcome space, characterizing the evolution of the distribution of discovered outcomes, i.e. the diversity of effects that the learner discovers how to produce. In order to characterize this exploration dynamics quantitatively, we monitored a measure which we refer to as *Kullback-Leibler Coverage* (KLC). At a given point in time during exploration, this measure computes the KL-divergence between the distribution of the outcomes produced so far, with a uniform distribution of outcomes in the space of physically possible outcomes (which is known by the experimenter, but unknown by the learner).

To compute it, we use a normalized histogram of the explored outcomes, with 30 bins per dimension, which we refer to as E , and we compute its Kullback Leibler Divergence with the normalized histogram of attainable points which we refer to as A :

$$KLC = \mathbb{D}_{KL}[E||A] = \sum_{i=1}^{30} E(i) \log \frac{E(i)}{A(i)}.$$

We emphasize that, when computed against a uniform distribution, the KLC measure is a proxy for the (opposite) Entropy of the E distribution. Nevertheless, we prefer to keep it under the divergence form, as the A distribution allows to define what the experimenter considers to be a good exploration distribution. In the case of this study, we consider a uniform distribution of explored locations over the attainable domain, to be the best exploration distribution achievable.

Baseline algorithms: We are using two natural baseline algorithms for evaluating the exploration dynamics of our IMGEP-UGL algorithmic implementations :

- **Random Goal Exploration with Engineered Features Representations (RGE-EFR):** This is an IMGEP implementation using a goal/outcome space with handcrafted features that directly encode the underlying structure of environments: for Arm-Ball, this is the 2D position of the ball in $[0, 1]^2$, and for Arm-Arrow this is the 2D position and the 1D orientation of the arrow in $[0, 1]^3$. This algorithm is also given the prior knowledge of $p(o) = \mathcal{U}(\mathcal{O})$. All other aspects of the IMGEP (regressor, meta-policy, other parameters) are identical to IMGEP-UGL implementations. This algorithm is known to provide highly efficient exploration dynamics in these environments (Forestier & Oudeyer, 2016).
- **Random Parameterization Exploration (RPE):** The Random Parameterization Exploration approach does not use an Outcome Space, nor a Goal Policy, and only samples a random parameterization $\theta \sim \mathcal{U}(\Theta)$ at each episode. We expected this algorithm to lower bound the performances of our novel architecture.

4 Results

We first study the exploration dynamics of all IMGEP-UGL algorithms, comparing them to the baselines and among themselves. Then, we study specifically the impact of the target embedding dimension (latent space) for the UGL implementations, by observing what exploration dynamics is produced in two cases:

- Using a target dimension larger than the true dimension ($l = 10$)
- Providing the true embedding dimension to the UGL implementations ($l = 2, 3$)

Finally, we specifically study RGE-VAE, using the intrinsic Gaussian prior of these algorithms to replace the *KDE* estimator of $p(O)$ in the UGL part.

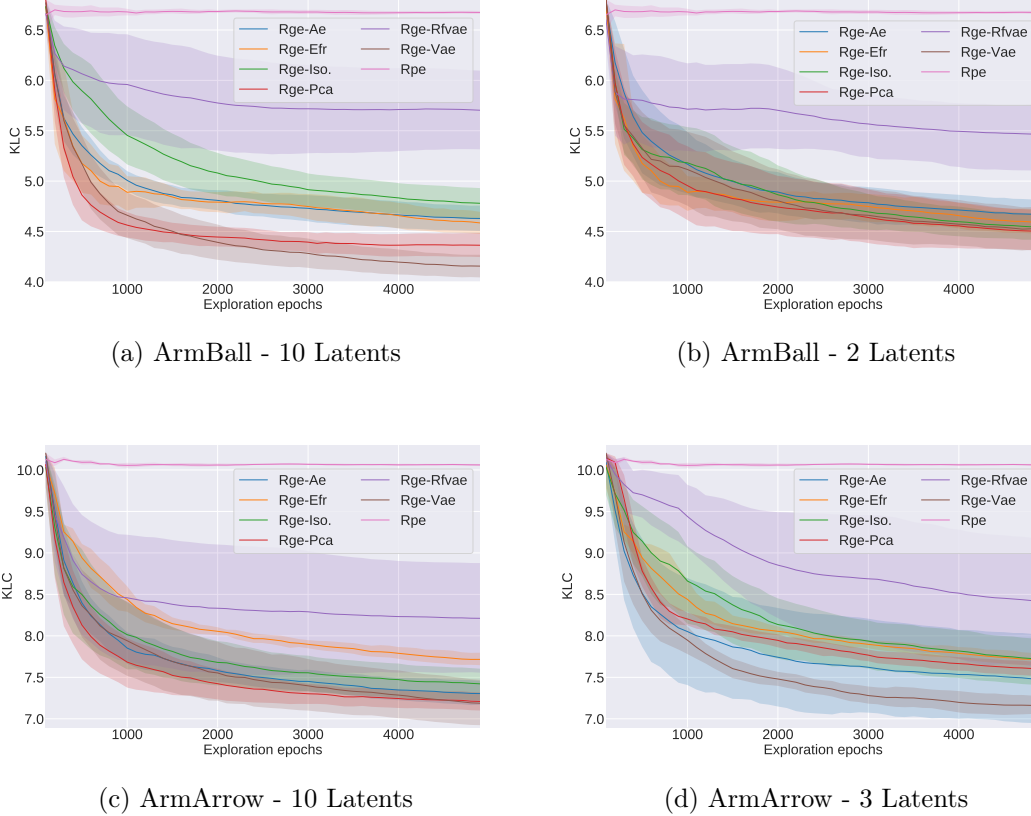
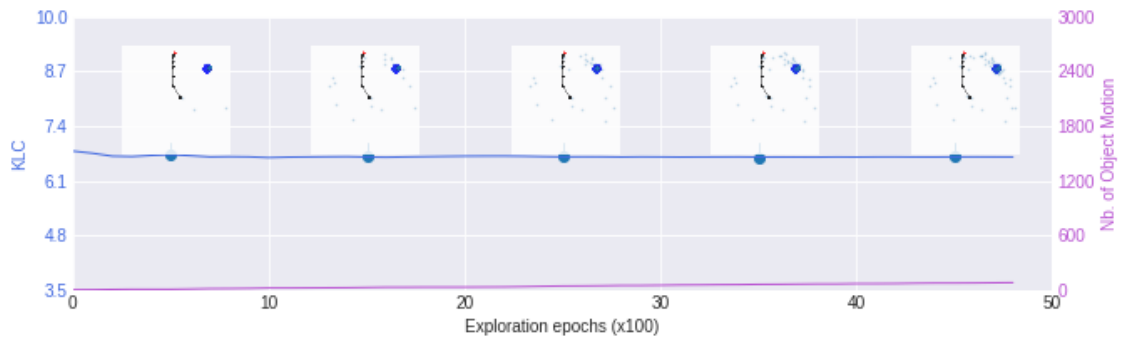


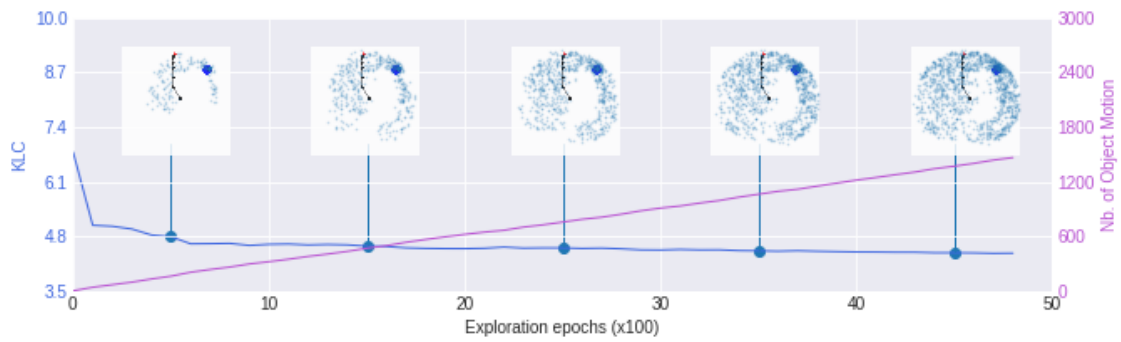
Figure 2: KL Coverage through epochs for different algorithms on ArmBall and ArmArrow environments. The exploration performance was assessed for both an over-complete representation (10 latent dimensions), and a complete representation (2 and 3 latent dimensions). The shaded area represent a 90% confidence interval estimated from 5 run of the different algorithms.

Exploration Performances: In Figure 2, we can see the evolution of the KLC through exploration epochs (one exploration epoch is defined as one experimentation / roll-out of a parameter θ). We can see that for both environments, and all values of latent spaces, all IMGEP-UGL algorithms, except RGE-RFVAE, achieve similar or better performance (both in terms of asymptotic KLC and speed to reach it) than the RGE-EFR algorithm using engineered Goal Space features, and much better performance than the RPE algorithm.

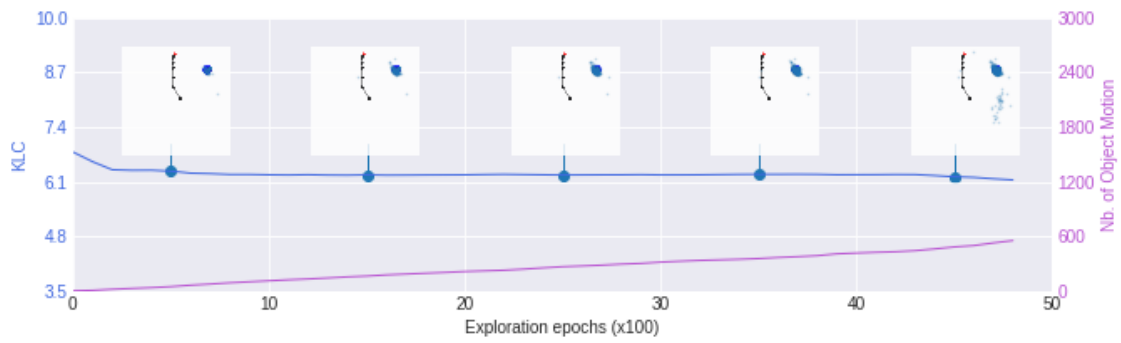
Figure 3 (see also Figure 8 and 9 in Appendix) show details of the evolution of discovered outcomes in ArmBall (final ball positions after the end of a policy roll-out) and corresponding KLC measures for individual runs with various algorithms. It also shows the evolution of the number of times learners managed to move the ball, which is considered in the KLC measure but not easily visible in the displayed



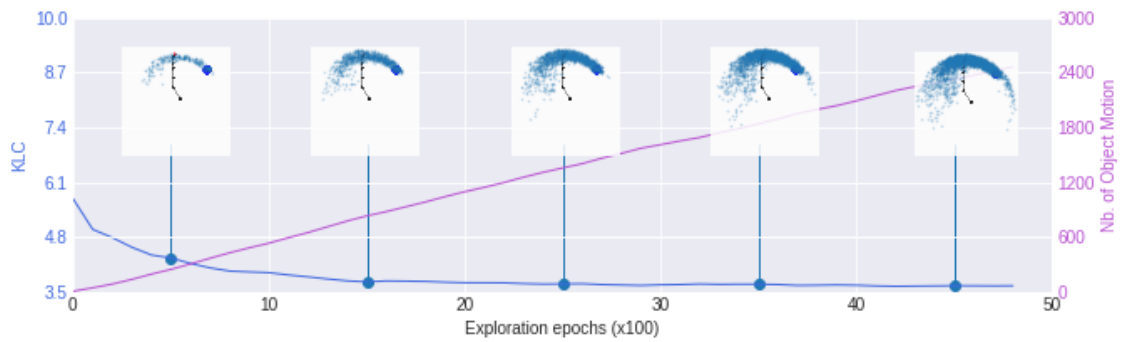
(a) Rpe



(b) Rge-Efr



(c) Rge-Rfvae - 10 Latents



(d) Rge-Vae - 10 Latents

Figure 3: Examples of achieved outcomes related with the evolution of KL-Coverage in the ArmBall environments. The number of times the ball was effectively handled is also represented.

set of outcomes in Figure 3. For instance, we observe that both RPE (Figure 3(a)) and RGE-RFVAE (Figure 3(c)) algorithms perform poorly: they discover very few policies moving the ball at all (pink curves), and these discovered ball moves cover only a small part of the physically possible outcome space. On the contrary, both RGE-EFR (handcrafted features) and RGE-VAE (learned goal space representation with VAE) perform very well, and the KLC of RGE-VAE is even better than the KLC of RGE-EFR, due to the fact that RGE-VAE has discovered more policies (around 2400) that move the ball than RGE-EFR (around 1600, pink curve).

Impact of target latent space size in IMGEP-UGL algorithms On the Arm-Ball problem, we observe that if one provides the true target embedding dimension ($l = 2$) to IMGEP-UGL implementations, RGE-Isomap is slightly improving (getting quasi-identical to RGE-EFR), RGE-AE does not change (remains quasi-identical to RGE-EFR), but the performance of RGE-PCA and RGE-VAE is degraded. For ArmArrow, the effect is similar: IMGEP-UGL algorithms with a larger target embedding dimension ($l = 10$) than the true dimensionality all perform better than RGE-EFR (except RGE-RFVAE which is worse in all cases), while when $l = 2$ only RGE-VAE is significantly better than RGE-EFR. In Appendix F, more examples of exploration curves with attached exploration scatters are shown. For most example runs, increasing the target embedding dimension enables learners to discover more policies moving the ball and, in these cases, the discovered outcomes are more concentrated towards the external boundary of the discus of physically possible outcomes. This behavior, where increasing the target embedding dimension improves the KLC while biasing the discovered outcome towards the boundary the feasible goals, can be understood as a consequence of the following well-known general property of IMGEPs: if goals are sampled outside the convex hull of outcomes already discovered, this has the side-effect of biasing exploration towards policies that will produce outcomes beyond this convex hull (until the boundary of feasible outcomes is reached). Here, as observations in the UGL phase were generated by uniformly moving the objects on the square $[-1, 1]^2$, while the feasible outcome space was the smaller discus of radius 1, goal sampling happened in a distribution of outcomes larger than the feasible outcome space. As one increases the embedding space dimensionality, the ratio between the volume of the corresponding hyper-cube and hyper-discus increases, in turn increasing the probability to sample goals outside the feasible space, which has the side effect of fostering the discovery of novel outcomes and biasing exploration towards the boundaries.

Impact of Sampling Kernel Density Estimation Another factor impacting the exploration assessed during our experiments was the importance of the distribution used as stationary Goal Policy. If, in most cases, the representation algorithm gives no particular prior knowledge of $p(o)$, in the case of Variational Auto-Encoders,

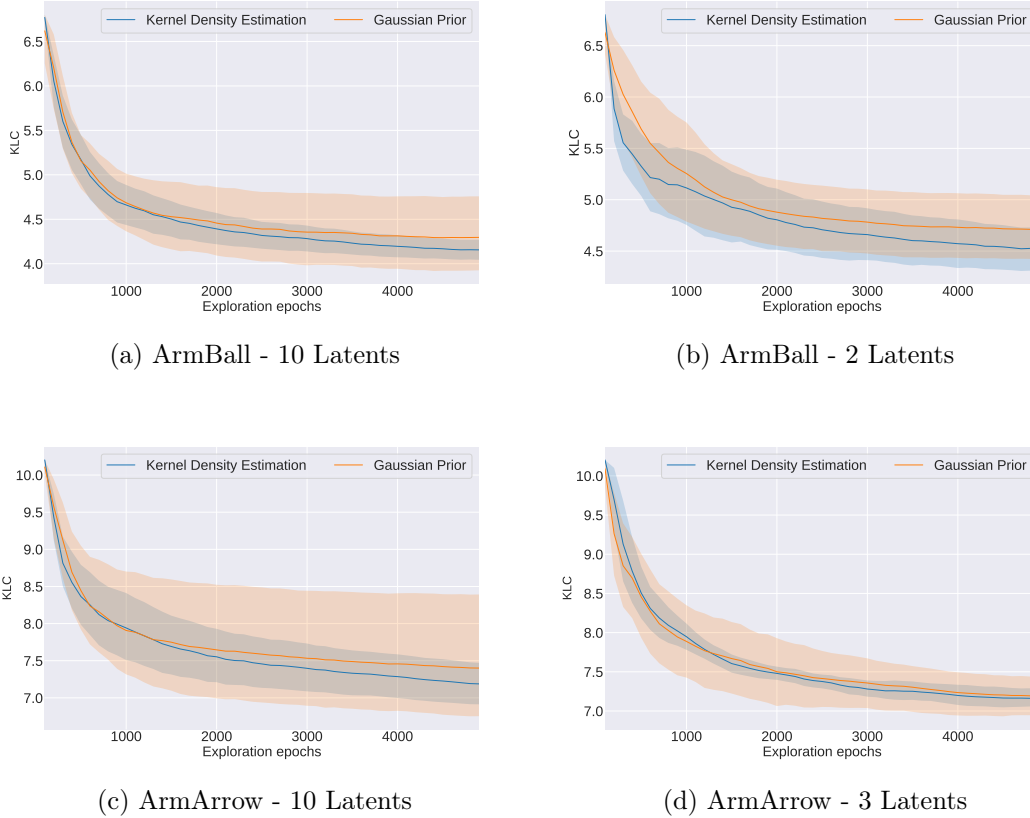


Figure 4: Evolution of the Exploration Ratio for RGE-VAE using KDE or Isotropic Gaussian prior. The curves show the mean and standard deviation over 5 independent runs of each condition.

it is assumed in the derivation that $p(o) = \mathcal{N}(0, I)$. Hence, the isotropic Gaussian distribution is a better candidate stationary Goal Policy than Kernel Density Estimation. Figure 4 shows a comparison between exploration performances achieved with RGE-VAE using a KDE distribution or an isotropic Gaussian as Goal Policy. The performance is not significantly different from the isotropic Gaussian case. Our experiments showed that convergence on the KL term of the loss can be more or less quick depending on the initialization. Since we used a number of iterations as stopping criterion for training (based on early experiments), we found that sometimes, at stop, the divergence was still pretty high despite achieving a low reconstruction error. In those cases the representation was not perfectly matching an isotropic Gaussian, which could lead to a goal sampling bias when using the isotropic Gaussian Goal Policy.

5 Conclusion

In this paper, we proposed a new Intrinsically Motivated Goal Exploration architecture with Unsupervised Learning of Goal spaces (IMGEP-UGL). Here, the Outcome Space (also used as Goal Space) representation is learned using passive observations of world changes through low-level raw sensors (e.g. movements of objects caused by another agent and perceived at the pixel level). Within the perspective of research on Intrinsically Motivated Goal Exploration started a decade ago (Oudeyer & Kaplan, 2007; Baranes & Oudeyer, 2013), and considering the fundamental problem of how AI agents can autonomously explore environments and skills by setting their own goals, this new architecture constitutes a milestone as it is to our knowledge the first goal exploration architecture where the goal space representation is learned, as opposed to hand-crafted.

Furthermore, we have shown in two simulated environments (involving a high-dimensional continuous action arm) that this new architecture can be successfully implemented using multiple kinds of unsupervised learning algorithms, including recent advanced deep neural network algorithms like Variational Auto-Encoders. This flexibility opens the possibility to benefit from future advances in unsupervised representation learning research. Yet, our experiments have shown that all algorithms we tried (except RGE-RFVAE) can compete with an IMGEP implementation using engineered feature representations. We also showed, in the context of our test environments, that providing to IMGEP-UGL algorithms a target embedding dimension larger than the true dimensionality of the phenomenon can be beneficial through leveraging exploration dynamics properties of IMGEPs. Though we must investigate more systematically the extent of this effect, this is encouraging from an autonomous learning perspective, as one should not assume that the learner initially knows the target dimensionality.

Limits and future work. The experiments presented here were limited to a fairly restricted set of environments. Experimenting over a larger set of environments would improve our understanding of IMGEP-UGL algorithms in general. In particular, a potential challenge is to consider environments where multiple objects/entities can be independently controlled, or where some objects/entities are not controllable (e.g. animate entities). In these cases, previous work on IMGEPs has shown that random Goal Policies should be either replaced by modular Goal Policies (considering a modular goal space representation, see Forestier et al. (2017)), or by active Goal Policies which adaptively focus the sampling of goals in subregions of the Goal Space where the competence progress is maximal (Baranes & Oudeyer, 2013). For learning modular representations of Goal Spaces, an interesting avenue of investigations could be the use of the Independently Controllable Factors approach proposed in (Thomas et al., 2017).

Finally, in this paper, we only studied a learning scenario where representation learning happens first in a passive perceptual learning stage, and is then fixed during

a second stage of autonomous goal exploration. While this was here motivated both by analogies to infant development and to facilitate evaluation, the ability to incrementally and jointly learn an outcome space representation and explore the world is a stimulating topic for future work.

Acknowledgement

This work was supported by Inria and by the European Commission, within the DREAM project, and has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement *N*^o 640891.

Bibliography

- Adolph, K. E., Bertenthal, B. I., Boker, S. M., Goldfield, E. C., and Gibson, E. J. (1997). Learning in the development of infant locomotion. *Monographs of the society for research in child development*, pages i–162.
- Albert, R. R., Schwade, J. A., and Goldstein, M. H. (2018). The social functions of babbling: acoustic and contextual characteristics that facilitate maternal responsiveness. *Developmental science*, 21(5):e12641.
- Alcock, J. (1972). The evolution of the use of tools by feeding animals. *Evolution*, pages 464–473.
- Andreae, P. M. and Andreae, J. H. (1978). A teachable machine in the real world. *International Journal of Man-Machine Studies*, 10(3):301–312.
- Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, O. P., and Zaremba, W. (2017). Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058.
- Antunes, A., Saponaro, G., Dehban, A., Jamone, L., Ventura, R., Bernardino, A., and Santos-Victor, J. (2015). Robotic tool use and problem solving based on probabilistic planning and learned affordances. In *IROS 2015 Workshop on Learning Object Affordances: a fundamental step to allow prediction, planning and tool use?*
- Auersperg, A. M., Von Bayern, A. M., Gajdon, G. K., Huber, L., and Kacelnik, A. (2011). Flexibility in problem solving and tool use of kea and new caledonian crows in a multi access box paradigm. *PLoS One*, 6(6):e20231.
- Austin, J. T. and Vancouver, J. B. (1996). Goal constructs in psychology: Structure, process, and content. *Psychological bulletin*, 120(3):338.
- Bakker, B., Schmidhuber, J., et al. (2004). Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In *Proc. of the 8-th Conf. on Intelligent Autonomous Systems*, pages 438–445.
- Baldassarre, G. and Mirolli, M. (2013). *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer.
- Bambach, S., Crandall, D., Smith, L., and Yu, C. (2018). Toddler-inspired visual object learning. In *Advances in Neural Information Processing Systems*, pages 1201–1210.

- Baranes, A. and Oudeyer, P.-Y. (2009). R-iac: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, 1(3):155–169.
- Baranes, A. and Oudeyer, P.-Y. (2010a). Intrinsically motivated goal exploration for active motor learning in robots: A case study. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1766–1773. IEEE.
- Baranes, A. and Oudeyer, P.-Y. (2010b). Maturationally-constrained competence-based intrinsically motivated learning. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*. IEEE.
- Baranes, A. and Oudeyer, P.-Y. (2011). The interaction of maturational constraints and intrinsic motivations in active motor development. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–8. IEEE.
- Baranes, A. and Oudeyer, P.-Y. (2013). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1).
- Barrett, T. M., Davis, E. F., and Needham, A. (2007). Learning about tools in infancy. *Developmental psychology*, 43(2):352.
- Barto, A. G. (2013). Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pages 17–47. Springer.
- Bates, E., Carlson-Luden, V., and Bretherton, I. (1980). Perceptual aspects of tool using in infancy. *Infant Behavior and Development*, 3:127–140.
- Beck, B. B. (1980). *Animal tool behavior*. Garland STPM Pub.
- Beck, S. R., Apperly, I. A., Chappell, J., Guthrie, C., and Cutting, N. (2011). Making tools isn’t child’s play. *Cognition*, 119(2):301–306.
- Beck, S. R., Cutting, N., Apperly, I. A., Demery, Z., Iliffe, L., Rishi, S., and Chappell, J. (2014). Is tool-making knowledge robust over time and across problems? *Frontiers in psychology*, 5:1395.
- Begus, K., Gliga, T., and Southgate, V. (2014). Infants learn what they want to learn: responding to infant pointing leads to superior learning. *PloS one*, 9(10).
- Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479.

- Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM.
- Bentley-Condit, V. et al. (2010). Animal tool use: current definitions and an updated comprehensive catalog. *Behaviour*, 147(2):185–32A.
- Benureau, F. C. and Oudeyer, P.-Y. (2016). Behavioral diversity generation in autonomous exploration through reuse of past experience. *Frontiers in Robotics and AI*, 3:8.
- Berlyne, D. E. (1960). Conflict, arousal, and curiosity.
- Berlyne, D. E. (1966). Curiosity and exploration. *Science*, 153(3731):25–33.
- Berthouze, L., Bakker, P., and Kuniyoshi, Y. (1996). Learning of oculo-motor control: a prelude to robotic imitation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, volume 1, pages 376–381. IEEE.
- Berthouze, L., Shigematsu, Y., and Kuniyoshi, Y. (1998). Dynamic categorization of explorative behaviors for emergence of stable sensorimotor configurations. In *Proceedings of the International Conference on Simulation of Adaptive Behavior (SAB1998)*, pages 67–72.
- Bickerton, D. (1990). *Language and species*. University of Chicago Press.
- Billard, A. (1999). Imitation skills as a means to enhance learning of a synthetic proto-language in an autonomous robot. In *Proceedings of the AISB Symposium on Imitation in Animals and Artifacts*, number CONF.
- Bonawitz, E. B., van Schijndel, T. J., Friel, D., and Schulz, L. (2012). Children balance theories and evidence in exploration, explanation, and learning. *Cognitive psychology*, 64(4):215–234.
- Borghi, A. M., Scorolli, C., Caligiore, D., Baldassarre, G., and Tummolini, L. (2013). The embodied mind extended: using words as social tools. *Frontiers in psychology*, 4:214.
- Bottou, L. (1998). Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142.
- Bourgeois, K. S., Khawar, A. W., Neal, S. A., and Lockman, J. J. (2005). Infant manual exploration of objects, surfaces, and their interrelations. *Infancy*, 8(3):233–252.

- Bourlard, H. and Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291–294.
- Braud, R., Pitti, A., and Gaussier, P. (2017). A modular dynamic sensorimotor model for affordances learning, sequences planning, and tool-use. *IEEE Transactions on Cognitive and Developmental Systems*, 10(1):72–87.
- Breazeal, C. and Scassellati, B. (1998). Infant-like social interactions between a robot and a human caretaker. *Adaptive Behavior*, 8(1).
- Brooks, R. and Meltzoff, A. N. (2008). Infant gaze following and pointing predict accelerated vocabulary growth through two years of age: A longitudinal, growth curve modeling study. *Journal of child language*, 35(1):207–220.
- Brown, A. L. (1990). Domain-specific principles affect learning and transfer in children. *Cognitive science*, 14(1):107–133.
- Brown, S. and Sammut, C. (2012). Tool use and learning in robots. *Encyclopedia of the Sciences of Learning*, pages 3327–3330.
- Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.
- Cabi, S., Colmenarejo, S. G., Hoffman, M. W., Denil, M., Wang, Z., and De Freitas, N. (2017). The intentional unintentional agent: Learning to solve many continuous control tasks simultaneously. *arXiv preprint arXiv:1707.03300*.
- Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., Tani, J., Belpaeme, T., Sandini, G., Nori, F., et al. (2010). Integration of action and language knowledge: A roadmap for developmental robotics. *Autonomous Mental Development, IEEE Transactions on*, 2(3).
- Cangelosi, A., Schlesinger, M., and Smith, L. B. (2015). *Developmental robotics: From babies to robots*. MIT Press.
- Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2018). Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*.
- Carroll, S. B. (2005). Endless forms most beautiful: The new science of evo devo and the making of the animal kingdom.
- Chang, M. B., Ullman, T., Torralba, A., and Tenenbaum, J. B. (2016). A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*.

- Chen, Z., Siegler, R. S., and Daehler, M. W. (2000). Across the great divide: Bridging the gap between understanding of toddlers' and older children's thinking. *Monographs of the Society for Research in Child Development*, pages i–105.
- Chentanez, N., Barto, A. G., and Singh, S. P. (2005). Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288.
- Churchill, A. W. and Fernando, C. (2014). An evolutionary cognitive architecture made of a bag of networks. *Evolutionary Intelligence*, 7(3):169–182.
- Clerkin, E. M., Hart, E., Rehg, J. M., Yu, C., and Smith, L. B. (2017). Real-world visual statistics and infants' first-learned object names. *Phil. Trans. R. Soc. B*, 372(1711).
- Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610.
- Cognolato, M., Atzori, M., and Müller, H. (2018). Head-mounted eye gaze tracking devices: An overview of modern devices and recent advances. *Journal of rehabilitation and assistive technologies engineering*, 5:2055668318773991.
- Cohen, L. and Billard, A. (2018). Social babbling: The emergence of symbolic gestures and words. *Neural Networks*, 106:194–204.
- Colas, C., Sigaud, O., and Oudeyer, P.-Y. (2018a). Curious: Intrinsically motivated multi-task, multi-goal reinforcement learning. *arXiv preprint arXiv:1810.06284*.
- Colas, C., Sigaud, O., and Oudeyer, P.-Y. (2018b). GEP-PG: Decoupling exploration and exploitation in deep reinforcement learning algorithms. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1039–1048, Stockholm, Sweden. PMLR.
- Connolly, K. and Dalglish, M. (1989). The emergence of a tool-using skill in infancy. *Developmental Psychology*, 25(6):894.
- Cook, C., Goodman, N. D., and Schulz, L. E. (2011). Where science starts: Spontaneous experiments in preschoolers' exploratory play. *Cognition*, 120(3):341–349.
- Cox, R. F. and Smitsman, A. W. (2006). The planning of tool-to-object relations in young children. *Developmental Psychobiology: The Journal of the International Society for Developmental Psychobiology*, 48(2):178–186.

- Cuccu, G. and Gomez, F. (2011). When novelty is not enough. In *European Conference on the Applications of Evolutionary Computation*, pages 234–243. Springer.
- Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.
- Cully, A. and Demiris, Y. (2017). Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation*, 22(2):245–259.
- Curtiss, S. (1977). *Genie: a psycholinguistic study of a modern-day wild child*. Academic Press.
- Da Silva, B., Konidaris, G., and Barto, A. (2012). Learning parameterized skills. In *ICML*, pages 1679–1686.
- Dautenhahn, K. and Billard, A. (1999). Studying robot social cognition within a developmental psychology framework. In *1999 Third European Workshop on Advanced Mobile Robots (Eurobot'99). Proceedings (Cat. No. 99EX355)*, pages 187–194. IEEE.
- Day, H. I. et al. (1971). Intrinsic motivation: A new direction in education.
- Dayan, P. and Sejnowski, T. J. (1996). Exploration bonuses and dual control. *Machine Learning*, 25(1):5–22.
- Dickinson, A. and Balleine, B. (1994). Motivational control of goal-directed action. *Animal Learning & Behavior*, 22(1):1–18.
- DiMercurio, A. J., Connell, J. P., Clark, M., and Corbetta, D. (2018). A naturalistic observation of spontaneous touches to the body and environment in the first 2 months of life. *Frontiers in psychology*, 9:2613.
- Dominey, P. F., Mallet, A., and Yoshida, E. (2009). Real-time spoken-language programming for cooperative interaction with a humanoid apprentice. *International Journal of Humanoid Robotics*, 6(02):147–171.
- Dosovitskiy, A. and Koltun, V. (2016). Learning to act by predicting the future. *arXiv preprint arXiv:1611.01779*.
- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.
- Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. (2019). Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*.

- Eimas, P. D., Siqueland, E. R., Jusczyk, P., and Vigorito, J. (1971). Speech perception in infants. *Science*, 171(3968):303–306.
- Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.
- Entezari, N., Shiri, M. E., and Moradi, P. (2010). A local graph clustering algorithm for discovering subgoals in reinforcement learning. In *International Conference on Future Generation Communication and Networking*, pages 41–50. Springer.
- Esseily, R., Rat-Fischer, L., O’regan, K., and Fagard, J. (2013). Understanding the experimenter’s intention improves 16-month-olds’ observational learning of the use of a novel tool. *Cognitive Development*, 28(1):1–9.
- Esseily, R., Rat-Fischer, L., Somogyi, E., O’Regan, K. J., and Fagard, J. (2016). Humour production may enhance observational learning of a new tool-use action in 18-month-old infants. *Cognition and Emotion*, 30(4):817–825.
- Fabisch, A. and Metzen, J. H. (2014). Active contextual policy search. *The Journal of Machine Learning Research*, 15(1):3371–3399.
- Fagard, J. and Lockman, J. J. (2010). Change in imitation for object manipulation between 10 and 12 months of age. *Developmental Psychobiology: The Journal of the International Society for Developmental Psychobiology*, 52(1):90–99.
- Fagard, J. and Marks, A. (2000). Unimanual and bimanual tasks and the assessment of handedness in toddlers. *Developmental Science*, 3(2):137–147.
- Fagard, J., Rat-Fischer, L., Esseily, R., Somogyi, E., and O’Regan, J. (2016). What does it take for an infant to learn how to use a tool by observation? *Frontiers in psychology*, 7:267.
- Fenson, L., Dale, P. S., Reznick, J. S., Bates, E., Thal, D. J., Pethick, S. J., Tomasello, M., Mervis, C. B., and Stiles, J. (1994). Variability in early communicative development. *Monographs of the society for research in child development*.
- Fernald, A., Taeschner, T., Dunn, J., Papousek, M., de Boysson-Bardies, B., and Fukui, I. (1989). A cross-language study of prosodic modifications in mothers’ and fathers’ speech to preverbal infants. *Journal of child language*, 16(3):477–501.
- Fernandes, J. J. R., Shahnazian, D., Holroyd, C. B., and Botvinick, M. M. (2018). Subgoal-and goal-related prediction errors in medial prefrontal cortex. *BioRxiv*, page 245829.
- Florensa, C., Held, D., Wulfmeier, M., Zhang, M., and Abbeel, P. (2017). Reverse curriculum generation for reinforcement learning. *arXiv preprint arXiv:1707.05300*.

- Fogassi, L., Ferrari, P. F., Gesierich, B., Rozzi, S., Chersi, F., and Rizzolatti, G. (2005). Parietal lobe: from action organization to intention understanding. *Science*, 308(5722):662–667.
- Fogel, A. and Hannan, T. E. (1985). Manual actions of nine-to fifteen-week-old human infants during face-to-face interaction with their mothers. *Child development*, pages 1271–1279.
- Forestier, S., Mollard, Y., and Oudeyer, P.-Y. (2017). Intrinsically motivated goal exploration processes with automatic curriculum learning. *arXiv preprint arXiv:1708.02190*.
- Forestier, S. and Oudeyer, P.-Y. (2016a). Curiosity-driven development of tool use precursors: a computational model. In *Proceedings of the 38th Annual Meeting of the Cognitive Science Society*.
- Forestier, S. and Oudeyer, P.-Y. (2016b). Modular active curiosity-driven discovery of tool use. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3965–3972.
- Forestier, S. and Oudeyer, P.-Y. (2016c). Overlapping waves in tool use development: a curiosity-driven computational model. In *Sixth Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*.
- Forestier, S. and Oudeyer, P.-Y. (2017). A unified model of speech and tool use early development. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*.
- Frank, M., Leitner, J., Stollenga, M., Förster, A., and Schmidhuber, J. (2014). Curiosity driven reinforcement learning for motion planning on humanoids. *Frontiers in neurorobotics*, 7:25.
- Franklin, B., Warlaumont, A. S., Messinger, D., Bene, E., Nathani Iyer, S., Lee, C.-C., Lambert, B., and Oller, D. K. (2014). Effects of parental interaction on infant vocalization rate, variability and vocal type. *Language Learning and Development*, 10(3):279–296.
- Friard, O. and Gamba, M. (2016). Boris: a free, versatile open-source event-logging software for video/audio coding and live observations. *Methods in Ecology and Evolution*, 7(11):1325–1330.
- Friston, K. J., Lin, M., Frith, C. D., Pezzulo, G., Hobson, J. A., and Ondobaka, S. (2017). Active inference, curiosity and insight. *Neural computation*, 29(10):2633–2683.

- Fromkin, V., Krashen, S., Curtiss, S., Rigler, D., and Rigler, M. (1974). The development of language in genie: A case of language acquisition beyond the “critical period”. *Brain and language*, 1(1):81–107.
- Gardner, R. A. and Gardner, B. T. (1969). Teaching sign language to a chimpanzee. *Science*, 165(3894):664–672.
- Gentilucci, M. (2003). Grasp observation influences speech production. *European Journal of Neuroscience*, 17(1):179–184.
- Gentilucci, M., Benuzzi, F., Gangitano, M., and Grimaldi, S. (2001). Grasp with hand and mouth: a kinematic study on healthy subjects. *Journal of Neurophysiology*, 86(4):1685–1699.
- Gibson, J. J. (1979). The theory of affordances. the ecological approach to visual perception.
- Gibson, K. R., Gibson, K. R., and Ingold, T. (1994). *Tools, language and cognition in human evolution*. Cambridge University Press.
- Goel, S. and Huber, M. (2003). Subgoal discovery for hierarchical reinforcement learning using learned policies. In *FLAIRS conference*, pages 346–350.
- Goldin-Meadow, S. (2007). Pointing sets the stage for learning language—and creating language. *Child development*, 78(3):741–745.
- Goldin-Meadow, S., Goodrich, W., Sauer, E., and Iverson, J. (2007). Young children use their hands to tell their mothers what to say. *Developmental science*, 10(6):778–785.
- Goldstein, M. H., King, A. P., and West, M. J. (2003). Social interaction shapes babbling: Testing parallels between birdsong and speech. *Proceedings of the National Academy of Sciences*, 100(13):8030–8035.
- Goldstein, M. H. and Schwade, J. A. (2008). Social feedback to infants’ babbling facilitates rapid phonological learning. *Psychological Science*, 19(5):515–523.
- Gonçalves, A., Abrantes, J., Saponaro, G., Jamone, L., and Bernardino, A. (2014). Learning intermediate object affordances: Towards the development of a tool concept. In *4th International Conference on Development and Learning and on Epigenetic Robotics*, pages 482–488. IEEE.
- Gopnik, A., Meltzoff, A. N., and Kuhl, P. K. (1999). *The scientist in the crib: Minds, brains, and how children learn*. William Morrow & Co.

- Gottlieb, J. and Balan, P. (2010). Attention as a decision in information space. *Trends in cognitive sciences*, 14(6):240–248.
- Gottlieb, J., Oudeyer, P.-Y., Lopes, M., and Baranes, A. (2013). Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in Cognitive Sciences*, 17(11).
- Graves, A., Bellemare, M. G., Menick, J., Munos, R., and Kavukcuoglu, K. (2017). Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1311–1320. JMLR.org.
- Greenfield, P. M. (1991). Language, tools and brain: The ontogeny and phylogeny of hierarchically organized sequential behavior. *Behavioral and brain sciences*, 14(4):531–551.
- Gregor, K., Rezende, D. J., and Wierstra, D. (2016). Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.
- Grizou, J., Points, L. J., Sharma, A., and Cronin, L. (2019). Exploration of self-propelling droplets using a curiosity driven robotic assistant. *arXiv preprint arXiv:1904.12635*.
- Gros-Louis, J., West, M. J., Goldstein, M. H., and King, A. P. (2006). Mothers provide differential feedback to infants’ prelinguistic sounds. *International Journal of Behavioral Development*, 30(6):509–516.
- Gros-Louis, J., West, M. J., and King, A. P. (2014). Maternal responsiveness and the development of directed vocalizing in social interactions. *Infancy*.
- Gruber, M. J., Gelman, B. D., and Ranganath, C. (2014). States of curiosity modulate hippocampus-dependent learning via the dopaminergic circuit. *Neuron*, 84(2):486–496.
- Guenther, F. H. (2006). Cortical interactions underlying the production of speech sounds. *Journal of communication disorders*, 39(5):350–365.
- Guenther, F. H. (2016). *Neural control of speech*. Mit Press.
- Guenther, F. H., Ghosh, S. S., and Tourville, J. A. (2006). Neural modeling and imaging of the cortical interactions underlying syllable production. *Brain and language*, 96(3).
- Guenther, F. H. and Vladusich, T. (2012). A neural theory of speech acquisition and production. *Journal of neurolinguistics*, 25(5):408–422.

- Guerin, F., Ferreira, P. A., and Indurkha, B. (2014). Using analogy to transfer manipulation skills. In *2014 AAAI Fall Symposium Series*.
- Guerin, F., Kruger, N., and Kraft, D. (2013). A survey of the ontogeny of tool use: from sensorimotor experience to planning. *Autonomous Mental Development, IEEE Transactions on*, 5(1).
- Hamilton, A. F. and Grafton, S. T. (2007). Action outcomes are represented in human inferior frontoparietal cortex. *Cerebral Cortex*, 18(5):1160–1168.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Hebb, D. O. (1955). Drives and the cns (conceptual nervous system). *Psychological review*, 62(4):243.
- Higgins, I., Matthey, L., Glorot, X., Pal, A., Uria, B., Blundell, C., Mohamed, S., and Lerchner, A. (2016). Early visual concept learning with unsupervised deep learning. *arXiv preprint arXiv:1606.05579*.
- Higuchi, S., Chaminade, T., Imamizu, H., and Kawato, M. (2009). Shared neural correlates for language and tool use in broca’s area. *Neuroreport*, 20(15):1376–1381.
- Hoicka, E., Bijvoet-van den Berg, S., Kerr, T., and Carberry, M. (2013). The unusual box test: A non-verbal, non-representational divergent thinking test for toddlers. In *2013 AAAI Spring Symposium Series*.
- Houthoofd, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. (2016). Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117.
- Howard, I. S. and Messum, P. (2011). Modeling the development of pronunciation in infant speech acquisition. *Motor Control*, 15(1).
- Howard, I. S. and Messum, P. (2014). Learning to pronounce first words in three languages: An investigation of caregiver and infant behavior using a computational model of an infant. *PloS One*, 9(10):e110334.
- Huang, X. and Weng, J. (2004). Motivational system for human-robot interaction. In *International Workshop on Computer Vision in Human-Computer Interaction*, pages 17–27. Springer.
- Hunt, J. (1965). Intrinsic motivation and its role in psychological development. In *Nebraska symposium on motivation*, volume 13, pages 189–282. University of Nebraska Press.

- Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373.
- Iriki, A., Tanaka, M., and Iwamura, Y. (1996). Coding of modified body schema during tool use by macaque postcentral neurones. *Neuroreport*, 7(14):2325–2330.
- Iriki, A. and Taoka, M. (2012). Triadic (ecological, neural, cognitive) niche construction: a scenario of human brain evolution extrapolating tool use and language from the control of reaching actions. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1585):10–23.
- Iverson, J. M. and Goldin-Meadow, S. (2005). Gesture paves the way for language development. *Psychological science*, 16(5):367–371.
- Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. (2016). Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*.
- Jepma, M., Verdonschot, R. G., Van Steenbergen, H., Rombouts, S. A., and Nieuwenhuis, S. (2012). Neural mechanisms underlying the induction and relief of perceptual curiosity. *Frontiers in behavioral neuroscience*, 6:5.
- Jonschkowski, R. and Brock, O. (2015). Learning state representations with robotic priors. *Autonomous Robots*, 39(3):407–428.
- Kaelbling, L. P. (1993). Learning to achieve goals. In *IJCAI*, pages 1094–1099. Citeseer.
- Kagan, J. (1972). Motives and development. *Journal of personality and social psychology*, 22(1):51.
- Kang, M. J., Hsu, M., Krajbich, I. M., Loewenstein, G., McClure, S. M., Wang, J. T.-y., and Camerer, C. F. (2009). The wick in the candle of learning: Epistemic curiosity activates reward circuitry and enhances memory. *Psychological Science*, 20(8):963–973.
- Kaplan, F. and Oudeyer, P.-Y. (2003). Motivational principles for visual know-how development.
- Kaplan, F. and Oudeyer, P.-Y. (2004). Maximizing learning progress: an internal reward system for development. In *Embodied artificial intelligence*, pages 259–270. Springer.
- Karmiloff-Smith, A. (1992). Beyond modularity: A developmental perspective on cognitive science.

- Kashdan, T. B., Stikma, M. C., Disabato, D. J., McKnight, P. E., Bekier, J., Kaji, J., and Lazarus, R. (2018). The five-dimensional curiosity scale: Capturing the bandwidth of curiosity and identifying four unique subgroups of curious people. *Journal of Research in Personality*, 73:130–149.
- Kenward, B., Rutz, C., Weir, A. A., and Kacelnik, A. (2006). Development of tool use in new caledonian crows: inherited action patterns and social influences. *Animal Behaviour*, 72(6):1329–1343.
- Kidd, C. and Hayden, B. Y. (2015). The psychology and neuroscience of curiosity. *Neuron*, 88(3):449–460.
- Kidd, C., Piantadosi, S. T., and Aslin, R. N. (2012). The goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PLoS One*, 7(5).
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Kompella, V. R., Stollenga, M., Luciw, M., and Schmidhuber, J. (2017). Continual curiosity-driven skill acquisition from high-dimensional video inputs for humanoid robots. *Artificial Intelligence*, 247:313–335.
- Koslowski, B. and Bruner, J. S. (1972). Learning to use a lever. *Child Development*, pages 790–799.
- Kozima, H. and Yano, H. (2001). A robot that learns to communicate with human caregivers. In *Proceedings of the First International Workshop on Epigenetic Robotics*, volume 2001.
- Kretchmar, R. M., Feil, T., and Bansal, R. (2003). Improved automatic discovery of subgoals for options in hierarchical. *Journal of Computer Science & Technology*, 3.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

- Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.
- Kuhl, P. K. (1991). Human adults and human infants show a "perceptual magnet effect" for the prototypes of speech categories, monkeys do not. *Perception & psychophysics*, 50(2):93–107.
- Kuhl, P. K. (2004). Early language acquisition: cracking the speech code. *Nature reviews neuroscience*, 5(11).
- Kuhl, P. K., Andruski, J. E., Chistovich, I. A., Chistovich, L. A., Kozhevnikova, E. V., Ryskina, V. L., Stolyarova, E. I., Sundberg, U., and Lacerda, F. (1997). Cross-language analysis of phonetic units in language addressed to infants. *Science*, 277(5326):684–686.
- Kuhl, P. K., Conboy, B. T., Coffey-Corina, S., Padden, D., Rivera-Gaxiola, M., and Nelson, T. (2007). Phonetic learning as a pathway to language: new data and native language magnet theory expanded (nlm-e). *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1493):979–1000.
- Kuhl, P. K., Conboy, B. T., Padden, D., Nelson, T., and Pruitt, J. (2005). Early speech perception and later language development: Implications for the "critical period". *Language learning and development*, 1(3-4):237–264.
- Kuhl, P. K. and Meltzoff, A. N. (1996). Infant vocalizations in response to speech: Vocal imitation and developmental change. *The journal of the Acoustical Society of America*, 100(4):2425–2438.
- Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 3675–3683.
- Kumaran, D., Hassabis, D., and McClelland, J. L. (2016). What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7):512–534.
- Kuniyoshi, Y., Yorozu, Y., Inaba, M., and Inoue, H. (2003). From visuo-motor self learning to early imitation—a neural architecture for humanoid learning. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 3, pages 3132–3139. IEEE.
- Kupcsik, A., Deisenroth, M. P., Peters, J., Loh, A. P., Vadakkepat, P., and Neumann, G. (2017). Model-based contextual policy search for data-efficient generalization of robot skills. *Artificial Intelligence*, 247:415–439.

- Kupcsik, A. G., Deisenroth, M. P., Peters, J., and Neumann, G. (2013). Data-efficient generalization of robot skills with contextual policy search. In *AAAI*.
- Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2016). Building machines that learn and think like people. *CoRR*, abs/1604.00289.
- Langevin, R. (1971). Is curiosity a unitary construct? *Canadian Journal of Psychology/Revue canadienne de psychologie*, 25(4):360.
- Lapeyre, M., Rouanet, P., Grizou, J., Nguyen, S., Depraetre, F., Le Falher, A., and Oudeyer, P.-Y. (2014). Poppy Project: Open-Source Fabrication of 3D Printed Humanoid Robot for Science, Education and Art. In *Digital Intelligence 2014*, Nantes, France.
- Laversanne-Finot, A., Pere, A., and Oudeyer, P.-Y. (2018). Curiosity driven exploration of learned disentangled goal spaces. In Billard, A., Dragan, A., Peters, J., and Morimoto, J., editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 487–504. PMLR.
- Lee, C.-C., Jhang, Y., Relyea, G., Chen, L.-m., and Oller, D. K. (2018). Babbling development as seen in canonical babbling ratios: A naturalistic evaluation of all-day recordings. *Infant Behavior and Development*, 50:140–153.
- Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.
- Lehman, J. and Stanley, K. O. (2011b). Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM.
- Li, W. and Fritz, M. (2015). Teaching robots the use of human tools from demonstration with non-dexterous end-effectors. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 547–553. IEEE.
- Little, D. Y.-J. and Sommer, F. T. (2013). Learning and exploration in action-perception loops. *Frontiers in neural circuits*, 7:37.
- Lockman, J. J. (2000). A perception–action perspective on tool use development. *Child development*, 71(1):137–144.
- Loewenstein, G. (1994). The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1):75.
- Lopes, M. and Oudeyer, P.-Y. (2012). The strategic student approach for life-long exploration and learning. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*.

- Lungarella, M. and Berthouze, L. (2003). Learning to bounce: first lessons from a bouncing robot. In *Proc. of the 2nd Int. Symp. on Adaptive Motion in Animals and Machines*.
- Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connection science*, 15(4):151–190.
- Mannor, S., Menache, I., Hoze, A., and Klein, U. (2004). Dynamic abstraction in reinforcement learning via clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 71. ACM.
- Mar, T., Tikhanoff, V., and Natale, L. (2018). What can i do with this tool? self-supervised learning of tool affordances from their 3-d geometry. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):595–610.
- Martius, G., Der, R., and Ay, N. (2013). Information driven self-organization of complex robotic behaviors. *PloS one*, 8(5):e63400.
- Massera, G., Tuci, E., Ferrauto, T., and Nolfi, S. (2010). The facilitatory role of linguistic instructions on developing manipulation skills. *IEEE Computational Intelligence Magazine*, 5(3):33–42.
- Matiisen, T., Oliver, A., Cohen, T., and Schulman, J. (2017). Teacher-student curriculum learning. *arXiv preprint arXiv:1707.00183*.
- McCall, R. B. and McGhee, P. E. (1977). The discrepancy hypothesis of attention and affect in infants. In *The structuring of experience*, pages 179–210. Springer.
- McGillion, M., Herbert, J. S., Pine, J., Vihman, M., DePaolis, R., Keren-Portnoy, T., and Matthews, D. (2017). What paves the way to conventional language? the predictive value of babble, pointing, and socioeconomic status. *Child development*, 88(1):156–166.
- McGovern, A. and Barto, A. G. (2001). Automatic discovery of subgoals in reinforcement learning using diverse density.
- Meltzoff, A. N. (1988). Imitation, objects, tools, and the rudiments of language in human ontogeny. *Human evolution*, 3(1-2):45–64.
- Meltzoff, A. N. and Warhol, J. (1999). Born to learn: What infants learn from watching us. *The role of early experience in infant development*, pages 145–164.
- Menache, I., Mannor, S., and Shimkin, N. (2002). Q-cut—dynamic discovery of sub-goals in reinforcement learning. In *European Conference on Machine Learning*, pages 295–306. Springer.

- Messum, P. and Howard, I. S. (2015). Creating the cognitive form of phonological units: The speech sound correspondence problem in infancy could be solved by mirrored vocal interactions rather than by imitation. *Journal of Phonetics*, 53:125–140.
- Metzen, J. H. and Kirchner, F. (2013). Incremental learning of skill collections based on intrinsic motivation. *Frontiers in Neurorobotics*, 7.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Morgan, T. J., Uomini, N. T., Rendell, L. E., Chouinard-Thuly, L., Street, S. E., Lewis, H. M., Cross, C. P., Evans, C., Kearney, R., de la Torre, I., et al. (2015). Experimental evidence for the co-evolution of hominin tool-making teaching and language. *Nature communications*, 6:6029.
- Moulin-Frier, C., Nguyen, S. M., and Oudeyer, P.-Y. (2013). Self-organization of early vocal development in infants and machines: the role of intrinsic motivation. *Frontiers in psychology*, 4.
- Moulin-Frier, C., Rouanet, P., Oudeyer, P.-Y., and others (2014). Explauto: an open-source Python library to study autonomous exploration in developmental robotics. In *ICDL-Epirob-International Conference on Development and Learning, Epirob*.
- Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.
- Mugan, J. and Kuipers, B. (2009). Autonomously learning an action hierarchy using a learned qualitative state representation.
- Mulcahy, N. J., Call, J., and Dunbar, R. I. (2005). Gorillas (*gorilla gorilla*) and orangutans (*pongo pygmaeus*) encode relevant problem features in a tool-using task. *Journal of comparative psychology*, 119(1):23.
- Nagai, Y., Asada, M., and Hosoda, K. (2002). Developmental learning model for joint attention. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 932–937. IEEE.
- Najnin, S. and Banerjee, B. (2017). A predictive coding framework for a developmental agent: Speech motor skill acquisition and speech production. *Speech Communication*, 92:24–41.
- Naylor, F. D. (1981). A state-trait curiosity inventory. *Australian Psychologist*, 16(2):172–183.

- Nguyen, S. M. and Oudeyer, P.-Y. (2012). Active choice of teachers, learning strategies and goals for a socially guided intrinsic motivation learner. *Paladyn*, 3(3):136–146.
- Nguyen, S. M. and Oudeyer, P.-Y. (2014). Socially guided intrinsic motivation for robot learning of motor skills. *Autonomous Robots*, 36(3):273–294.
- Nicholson, J. S., Deboeck, P. R., and Howard, W. (2017). Attrition in developmental psychology: A review of modern missing data reporting and practices. *International Journal of Behavioral Development*, 41(1):143–153.
- Oller, D., Levine, S. L., Cobo-Lewis, A. B., Eilers, R. E., and Pearson, B. Z. (1998). Vocal precursors to linguistic communication: How babbling is connected to meaningful speech. *Exploring the speech-language connection*, 8:1–25.
- Oller, D. K. (2000). *The emergence of the speech capacity*. Psychology Press.
- Oller, D. K., Dale, R., and Griebel, U. (2016). New frontiers in language evolution and development. *Topics in cognitive science*, 8(2):353–360.
- Örnkloo, H. and von Hofsten, C. (2007). Fitting objects into holes: On the development of spatial cognition skills. *Developmental Psychology*, 43(2):404.
- Osiurak, F., Morgado, N., and Palluel-Germain, R. (2012). Tool use and perceived distance: When unreachable becomes spontaneously reachable. *Experimental Brain Research*, 218(2):331–339.
- Oudeyer, P.-Y., Baranes, A., and Kaplan, F. (2013). Intrinsically motivated learning of real-world sensorimotor skills with developmental constraints. In *Intrinsically motivated learning in natural and artificial systems*, pages 303–365. Springer.
- Oudeyer, P.-Y., Gottlieb, J., and Lopes, M. (2016). Intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies. *Progress in brain research*, 229:257–284.
- Oudeyer, P.-Y. and Kaplan, F. (2007). What is intrinsic motivation? A typology of computational approaches. *Frontiers in Neurorobotics*, 1.
- Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. (2007). Intrinsic Motivation Systems for Autonomous Mental Development. *IEEE Transactions on Evolutionary Computation*, 11(2).
- Oudeyer, P.-Y. and Smith, L. B. (2016). How evolution may work through curiosity-driven developmental process. *Topics in Cognitive Science*, 8(2):492–502.
- Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

- Passingham, R. E. and Wise, S. P. (2012). *The neurobiology of the prefrontal cortex: anatomy, evolution, and the origin of insight*. Number 50. Oxford University Press.
- Pastra, K. and Aloimonos, Y. (2012). The minimalist grammar of action. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1585):103–117.
- Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. *arXiv preprint arXiv:1705.05363*.
- Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.
- Penney, R. and McCann, B. (1964). The children’s reactive curiosity scale. *Psychological reports*, 15(1):323–334.
- Pfeifer, R. and Scheier, C. (1997). Sensory—motor coordination: The metaphor and beyond. *Robotics and autonomous systems*, 20(2-4):157–178.
- Philippsen, A., Reinhart, F., and Wrede, B. (2015). Efficient bootstrapping of vocalization skills using active goal babbling.
- Philippsen, A. K. (2018). Learning how to speak. goal space exploration for articulatory skill acquisition.
- Philippsen, A. K., Reinhart, R. F., and Wrede, B. (2014). Learning how to speak: Imitation-based refinement of syllable production in an articulatory-acoustic model. In *4th International Conference on Development and Learning and on Epigenetic Robotics*, pages 195–200. IEEE.
- Piaget, J. (1952). *The origins of intelligence in children*. International Universities Press New York.
- Power, T. G. (1999). *Play and exploration in children and animals*. Psychology Press.
- Péré, A., Forestier, S., Sigaud, O., and Oudeyer, P.-Y. (2018). Unsupervised learning of goal spaces for intrinsically motivated goal exploration. In *International Conference on Learning Representations*.
- Queißer, J. F., Reinhart, R. F., and Steil, J. J. (2016). Incremental bootstrapping of parameterized motor skills. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 223–229. IEEE.
- Räsänen, O. (2012). Computational modeling of phonetic and lexical learning in early language acquisition: Existing models and future directions. *Speech Communication*, 54(9):975–997.

- Rat-Fischer, L., O'Regan, J. K., and Fagard, J. (2012). The emergence of tool use during the second year of life. *Journal of experimental child psychology*, 113(3):440–446.
- Reinke, C., Etcheverry, M., and Oudeyer, P.-Y. (2019). Intrinsically motivated exploration for automated discovery of patterns in morphogenetic systems. *arXiv preprint arXiv:1908.06663*.
- Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.
- Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degraeve, J., Van de Wiele, T., Mnih, V., Heess, N., and Springenberg, J. T. (2018). Learning by playing-solving sparse reward tasks from scratch. *arXiv preprint arXiv:1802.10567*.
- Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annu. Rev. Neurosci.*, 27:169–192.
- Rolf, M., Steil, J., and Gienger, M. (2010). Goal babbling permits direct learning of inverse kinematics. *IEEE Transactions on Autonomous Mental Development*, 2(3).
- Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837.
- Rowe, M. L., Özçalışkan, Ş., and Goldin-Meadow, S. (2008). Learning words by hand: Gesture's role in predicting vocabulary development. *First language*, 28(2):182–199.
- Roy, D. K. (2002). Learning visually grounded words and syntax for a scene description task. *Computer speech & language*, 16(3-4):353–385.
- Ryan, R. M. and Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67.
- Sakai, K. L. (2005). Language acquisition and brain development. *Science*, 310(5749):815–819.
- Salge, C., Glackin, C., and Polani, D. (2014). Changing the environment based on empowerment as intrinsic motivation. *Entropy*, 16(5):2789–2819.
- Santucci, V. G., Baldassarre, G., and Mirolli, M. (2013). Which is the best intrinsic motivation signal for learning multiple skills? *Frontiers in neurorobotics*, 7:22.

- Sasaki, C. T., Levine, P. A., Laitman, J. T., and Crelin, E. S. (1977). Postnatal descent of the epiglottis in man: a preliminary report. *Archives of Otolaryngology*, 103(3):169–171.
- Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):537–547.
- Schauble, L. (1996). The development of scientific reasoning in knowledge-rich contexts. *Developmental Psychology*, 32(1):102.
- Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *International conference on machine learning*, pages 1312–1320.
- Schmerling, M., Schillaci, G., and Hafner, V. V. (2015). Goal-directed learning of hand-eye coordination in a humanoid robot. In *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 168–175. IEEE.
- Schmidhuber, J. (1991a). Curious model-building control systems. In *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, pages 1458–1463. IEEE.
- Schmidhuber, J. (1991b). A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pages 222–227.
- Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3).
- Schmidhuber, J. (2013). Powerplay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. *Frontiers in psychology*, 4:313.
- Schulz, L. E. and Bonawitz, E. B. (2007). Serious fun: preschoolers engage in more exploratory play when evidence is confounded. *Developmental psychology*, 43(4):1045.
- Scott, D. W. (2015). *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.
- Seed, A. and Byrne, R. (2010). Animal tool-use. *Current biology*, 20(23):R1032–R1039.
- Selfridge, O. G. (1993). The gardens of learning: a vision for ai. *AI Magazine*, 14(2):36–36.

- Shrager, J. and Siegler, R. S. (1998). Scads: A model of children’s strategy choices and strategy discoveries. *Psychological Science*, 9(5):405–410.
- Siegler, R. S. (1996). *Emerging minds: The process of change in children’s thinking*. Oxford University Press.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.
- Şimşek, Ö. and Barto, A. G. (2004). Using relative novelty to identify useful temporal abstractions in reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 95. ACM.
- Şimşek, Ö. and Barto, A. G. (2009). Skill characterization based on betweenness. In *Advances in neural information processing systems*, pages 1497–1504.
- Şimşek, Ö., Wolfe, A. P., and Barto, A. G. (2005). Identifying useful subgoals in reinforcement learning by local graph partitioning. In *Proceedings of the 22nd international conference on Machine learning*, pages 816–823. ACM.
- Singh, S., Lewis, R. L., Barto, A. G., and Sorg, J. (2010). Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82.
- Skinner, B. F. (1953). *Science and human behavior*. Number 92904. Simon and Schuster.
- Slaughter, V. and Suddendorf, T. (2007). Participant loss due to “fussiness” in infant visual paradigms: A review of the last 20 years. *Infant Behavior and Development*, 30(3):505–514.
- Somogyi, E., Ara, C., Gianni, E., Rat-Fischer, L., Fattori, P., O’Regan, J. K., and Fagard, J. (2015). The roles of observation and manipulation in learning to use a tool. *Cognitive Development*, 35:186–200.
- Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. (2016). Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746.
- Srivastava, R. K., Steunebrink, B. R., and Schmidhuber, J. (2013). First experiments with powerplay. *Neural Networks*, 41:130–136.
- Stanley, K. O. and Lehman, J. (2015). *Why greatness cannot be planned: The myth of the objective*. Springer.

- Stout, A. and Barto, A. G. (2010). Competence progress intrinsic motivation. In *2010 IEEE 9th International Conference on Development and Learning*, pages 257–262. IEEE.
- Stoytchev, A. (2005). Behavior-grounded representation of tool affordances. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE.
- Strehl, A. L. and Littman, M. L. (2008). An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331.
- Stulp, F., Herlant, L., Hoarau, A., and Raiola, G. (2014). Simultaneous on-line discovery and improvement of robotic skill options. In *International Conference on Intelligent Robots and Systems (IROS)*.
- Stulp, F., Raiola, G., Hoarau, A., Ivaldi, S., and Sigaud, O. (2013). Learning compact parameterized skills with a single regression. In *IEEE-RAS International Conference on Humanoid Robots*.
- Stulp, F. and Sigaud, O. (2013). Robot skill learning: From reinforcement learning to evolution strategies. *Paladyn. Journal of Behavioral Robotics*, 4(1):49–61.
- Sugita, Y. and Tani, J. (2005). Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive behavior*, 13(1):33–52.
- Sutskever, I. and Zaremba, W. (2014). Learning to execute. *arXiv preprint arXiv:1410.4615*.
- Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the seventh international conference on machine learning*, pages 216–224.
- Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. (2011). Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation for Autonomous Agents and Multiagent Systems.
- Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.
- Tang, H., Houthoofd, R., Foote, D., Stooke, A., Chen, O. X., Duan, Y., Schulman, J., DeTurck, F., and Abbeel, P. (2017). # exploration: A study of count-based

- exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pages 2753–2762.
- Taylor, A. H., Hunt, G. R., Holzhaider, J. C., and Gray, R. D. (2007). Spontaneous metatool use by new caledonian crows. *Current Biology*, 17(17):1504–1507.
- Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.
- Thelen, E. (1979). Rhythmical stereotypies in normal human infants. *Animal behaviour*, 27:699–715.
- Thelen, E. and Smith, L. B. (1996). *A dynamic systems approach to the development of cognition and action*. MIT press.
- Thill, S., Svensson, H., and Ziemke, T. (2011). Modeling the development of goal-specificity in mirror neurons. *Cognitive computation*, 3(4):525–538.
- Thill, S. and Twomey, K. E. (2016). What’s on the inside counts: A grounded account of concept acquisition and development. *Frontiers in psychology*, 7.
- Thomas, V., Pondard, J., Bengio, E., Sarfati, M., Beaudoin, P., Meurs, M.-J., Pineau, J., Precup, D., and Bengio, Y. (2017). Independently controllable factors. *arXiv preprint arXiv:1708.01289*.
- Tikhanoff, V., Cangelosi, A., and Metta, G. (2010). Integration of speech and action in humanoid robots: icub simulation experiments. *IEEE Transactions on Autonomous Mental Development*, 3(1):17–29.
- Tikhanoff, V., Pattacini, U., Natale, L., and Metta, G. (2013). Exploring affordances and tool use on the icub. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 130–137. IEEE.
- Tomasello, M., Carpenter, M., and Liszkowski, U. (2007). A new look at infant pointing. *Child development*, 78(3):705–722.
- Tomczak, J. M. and Welling, M. (2016). Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*.
- Turing, A. (1950). *Computing machinery and intelligence*. *Mind*, 59, 433–460.
- Ude, A., Gams, A., Asfour, T., and Morimoto, J. (2010). Task-specific generalization of discrete and periodic dynamic movement primitives. *Robotics, IEEE Transactions on*, 26(5):800–815.

- Ugur, E., Nagai, Y., Sahin, E., and Oztop, E. (2015). Staged development of robot skills: Behavior formation, affordance learning and imitation with motionese. *IEEE Transactions on Autonomous Mental Development*, 7(2):119–139.
- Ugur, E. and Piater, J. (2016). Emergent structuring of interdependent affordance learning tasks using intrinsic motivation and empirical feature selection. *IEEE Transactions on Cognitive and Developmental Systems*, 9(4):328–340.
- Uzgiris, I. C. and Hunt, J. (1975). Assessment in infancy: Ordinal scales of psychological development.
- Van Dijk, S. G. and Polani, D. (2011). Grounding subgoals in information transitions. In *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 105–111. IEEE.
- Van Lawick-Goodall, J. (1971). Tool-using in primates and other vertebrates. In *Advances in the Study of Behavior*, volume 3, pages 195–249. Elsevier.
- Van Leeuwen, L., Smitsman, A., and van Leeuwen, C. (1994). Affordances, perceptual complexity, and the development of tool use. *Journal of Experimental Psychology: Human Perception and Performance*, 20(1):174.
- Vigorito, C. M. and Barto, A. G. (2010). Intrinsically motivated hierarchical skill learning in structured environments. *Autonomous Mental Development, IEEE Transactions on*, 2(2).
- Von Hofsten, C. (2004). An action perspective on motor development. *Trends in cognitive sciences*, 8(6):266–272.
- Voss, H.-G. and Keller, H. (1983). *Curiosity and exploration: Theories and results*. Academic Press.
- Warlaumont, A. S. (2013). Saliency-based reinforcement of a spiking neural network leads to increased syllable production. In *Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on*, pages 1–7. IEEE.
- Warlaumont, A. S., Westermann, G., Buder, E. H., and Oller, D. K. (2013). Prespeech motor learning in a neural network using reinforcement. *Neural Networks*, 38:64–75.
- White, R. W. (1959). Motivation reconsidered: The concept of competence. *Psychological review*, 66(5):297.
- Wicaksono, H. and Sammut, C. (2015). A learning framework for tool creation by a robot.

- Wiering, M. and Schmidhuber, J. (1996). Hq-learning: Discovering markovian subgoals for non-markovian reinforcement learning. *Technical report IDSIA-95-96*, pages 1–13.
- Willatts, P. (1990). Development of problem-solving strategies in infancy. *Children's strategies: Contemporary views of cognitive development*, pages 23–66.
- Willatts, P. (1999). Development of means–end behavior in young infants: Pulling a support to retrieve a distant object. *Developmental psychology*, 35(3):651.
- Williams, J. L., Corbetta, D., and Cobb, L. (2015). How perception, action, functional value, and context can shape the development of infant reaching. *Movement & Sport Sciences-Science & Motricité*, (89):5–15.
- Wimpenny, J. H., Weir, A. A., Clayton, L., Rutz, C., and Kacelnik, A. (2009). Cognitive processes associated with sequential tool use in new caledonian crows. *PLoS One*, 4(8):e6471.
- Zaremba, W. and Sutskever, I. (2014). Learning to execute. *arXiv preprint arXiv:1410.4615*.
- Zelazo, P. R. and Kearsley, R. B. (1980). The emergence of functional play in infants: Evidence for a major cognitive transition. *Journal of Applied Developmental Psychology*, 1(2).
- Zlatin, M. A. (1975). Explorative mapping of the vocal tract and primitive syllabification in infancy: The first six months. *Purdue University Contributed Papers. Speech... Hearing... Language*, (5):58–73.