Agents artificiels autotelic et sociaux : formation et exploitation de conventions culturelles chez les agents artificiels autonomes incarnés

Tristan Karch

Towards Social Autotelic Artificial Agents

Formation and Exploitation of Cultural Conventions in Autonomous Embodied Artificial Agents

---

By **Tristan KARCH**

Under the supervision of Pierre-Yves OUDEYER & Clément MOULIN-FRIER

In partial fulfillment of the requirements

for the degree of Doctor of Philosophy

University of Bordeaux

Graduate school of Mathematics and Computer Science

Major in Computer Science

Submitted on March 24, 2023. Defended on May 11, 2023.

Composition of the jury:

Pr. John LANGFORD	Research Director	Microsoft Research	Reviewer
Pr. Noah GOODMAN	Associate Professor	Stanford University	Reviewer
Pr. Stefano PALMINTERI	Research Director	INSERM & ENS	Examinator
Dr. Andrew LAMPINEN	Senior Research Scientist	Deepmind	Examinator
Dr. Xavier HINAUT	Researcher	INRIA	Examinator & President
Dr. Pierre-Yves OUDEYER	Research Director	INRIA	Director
Dr. Clément MOULIN-FRIER	Researcher	INRIA	Director

Contents

Acknowledgments

Over the course of the last three years, I've been surrounded by amazing people who I would like to thank here, as I could never have completed this research without them.

First, my deepest gratitude goes to my supervisors Clément Moulin-Frier and Pierre-Yves Oudeyer for their invaluable support throughout my research journey. They provided me with incredible guidance and advice in the early stages of my projects but were also always available at the conclusion of each of them (including this manuscript).

I would like to warmly thank Prof. Emmanuel Rachelson from ISAE-Supaero in Toulouse who trusted me and welcomed me into his group during my Engineering Diploma which really got me prepared for my research journey.

I have had the privilege of working with brilliant colleagues in the Flowers Lab at INRIA Bordeaux. It was amazing to share such a fantastic working environment with all the members of the team. I am especially grateful for the personal and scientific discussions with Cédric Colas, Laetitia Teodorescu, Remy Portelas, Benjamin Clément, Mayalen Etcheverry, Eleni Nisioti, Clément Romac, Gauthier Hamon, Yoann Lemesle and Thomas Carta, among others.

I'd like to thank all my co-authors who helped me publish exciting projects: Pierre-Yves Oudeyer, Clément Moulin-Frier, Olivier Sigaud, Katja Hoffman, Derek Nowrouzezahrai, Christopher Pal, Peter Ford-Dominey, Romain Laroche, Nicolas Lair, Yoann Lemesle, Laetitia Teodorescu, Paul Barde, and Cédric Colas.

I am particularly grateful to Cédric Colas who acted as a mentor at the beginning of my thesis, providing me with all the necessary tools to carry out efficient research and with whom I carry out many projects.

I am also very happy to be able to count on my good old friend Paul Barde who helped confirm my decision to pursue a doctoral degree and with whom I can talk about everything.

Now that we move on to friends, I would like to thank them all. It is impossible to list them here, but, anyway, they will probably never read these lines. If they somehow land here, they will recognize themselves from the cities in which we lived together (Pays-de-Gex, Lausanne, Copenhagen, Toulouse, and New York).

Finally my biggest thanks go to my family and particularly to my girlfriend, Sanoé Leroux, who had to adapt to my weird working hours and who always supported me.

Abstract

One of the fundamental goals of Artificial Intelligence (AI) is to design embodied autonomous agents that can evolve in various environments, perform a multitude of tasks and interact with humans. To this end, AI researchers employ various approaches, with two primary methods standing out: developmental robotics and standard AI paradigms. While developmental robotics models agents' cognitive development in simplified environments, standard AI paradigms focus on algorithmic contributions in precise and technical benchmarks. In this thesis, we extend upon recent calls to bridge these two fields and investigate the role of cultural conventions in the development of artificial agents using state-of-the-art AI algorithms.

This research leverages work from developmental psychology and focuses on two crucial aspects of human development, namely autotelic and social learning. The former enables agents to form open-ended repertoires of skills by inventing and pursuing their own goals while the latter enables them to communicate, cooperate, teach, and organize their thoughts. Our contributions are organized around two fundamental scientific questions: 1) the formation of cultural conventions within populations of artificial agents, and 2) the exploitation of cultural conventions in their cognitive development.

The first part of this manuscript deals with the formation of cultural conventions. It builds on recent studies in the field of emergent communication to propose two computational studies. The first one investigates the formation of cultural conventions in the ecological context where artificial agents communicate via a graphical sensory-motor channel. The second one draws inspiration from experimental semiotics and studies the emergence of communication in the architect-builder problem: a novel interactive learning paradigm where agents have asymmetries of information and affordances which makes the application of standard Multi-Agent Reinforcement Learning impossible.

The second part focuses on the exploitation of cultural conventions. Inspired by the pioneering work of Vygotsky and other psychologists we first introduce the Vygotskian Autotelic AI Framework. This framework enables Reinforcement Learning agents to internalize social interactions in order to transform their cognitive abilities enabling them to form abstract representations, achieve systematic generalization, and creatively explore their environment. Following this conceptual contribution, we propose two computational studies. The first one explores the role of inductive biases in the language grounding problem where agents need to align their physical experience of the world with linguistic inputs provided by social partners. Our final computational contribution introduces the IMAGINE agent: a Vygotskian autotelic agent that converts linguistic descriptions given by a social partner into targetable goals. IMAGINE leverages language productivity and systematic generalization to grow an open-ended repertoire of skills in a creative way.

Glossary

Action-value Function The action-value function $Q_\pi(s, a)$ is the expected return of the trajectory taking action a from state s before following π from the next state s'. 13

Autotelic from the Greek *auto* (self) and *telos* (end, goal), characterizes agents that generate their own goals and learning signals. In is equivalent to *intrinsically motivated and goal-conditioned*. 3

Compositionality our ability to understand language and create meaningful expressions by assembling other meaningful expressions. 30

Cultural Convention any social production, linguistic or physical, internal or interpersonal, used to communicate, cooperate, teach, think, or transmit. 5

Developmental Artificial Intelligence a multidisciplinary field that integrates principles from artificial intelligence, developmental psychology, and neuroscience to simulate and analyze the cognitive mechanisms of artificial agents. 25

Goal a $g = (z_g, R_g)$ pair where z_g is a compact *goal parameterization* or *goal embedding* and R_g is a *goal-achievement* function. 19

Goal-achievement function $R_g(\cdot) = R_\mathcal{G}(\cdot \mid z_g)$ where $R_\mathcal{G}$ is a goal-conditioned reward function.. 19

Goal-conditioned policy a function that generates the next action given the current state and the goal. 19

Markov Decision Process A Markov Decision Process (MDP) models a decision-making problem using a set of states, a set of actions, and a set of probabilities that describe the outcome of each action in each state. 12

Markov Game The framework of Markov Games is a multi-agent extension of MDPs. 23

Open-ended learning developmental kind of learning where the objectives are not predetermined, but rather the learner is encouraged to discover knowledge and skills through an open exploration process. 27

Self-organization process by which spontaneously ordered patterns and structures emerge in a system without the need for central control or external guidance. 27

Skill the association of a goal and a policy to reach it. 19

Value Function The value function $V_\pi(s)$ of a policy π gives the expected return of a trajectory starting from s and following π. 13

Chapter 1

Introduction

One fundamental goal of Artificial Intelligence (AI) is to design embodied autonomous interactive agents that can evolve in various environments and complete a wide range of tasks. To that end, researchers in AI take several angles of attack and rely on different paradigms that consider different drivers for learning. In Reinforcement Learning (RL) (Sutton & Barto, 2018), agents learn from *exploration* of their environment. They rely solely on their experience of the world in order to solve a pre-defined task. In Imitation Learning (IL) (Pomerleau, 1991), agents learn from *demonstrations*, i.e. trajectories provided by an expert that correspond to the transitions required to take to solve a pre-defined task. In Multi-Agent Reinforcement Learning (MARL) (Littman, 1994), agents learn in *cooperation* and need to interact with each other in order to solve collaborative tasks.

Recent extensions of RL algorithm have shown success in solving a wealth of problems such as playing the Atari videogames at super-human levels (Mnih et al., 2015), beating chess and go world champions (Silver et al., 2016), controlling stratospheric baloons (Bellemare et al., 2020) or even maintaining plasma in fusion reactors (Degrave et al., 2022). Similarly, IL methods coupled to Transformers (Vaswani et al., 2017) have enabled the training of a generalist agent on a massive dataset of diverse interactions (Reed et al., 2022). It has also been used to perform in-context reinforcement learning via algorithm distillation (Laskin et al., 2022). Finally, multi-agent methods have permitted populations of agents to play hide and seek (Baker et al., 2020) or even to collaboratively solve common-pool resource problems (Pérolat et al., 2017).

But unlike humans, these algorithms are still heavily sample-inefficient, requiring billions of transitions to become proficient on isolated tasks. Most importantly, they lack the ability to generalize and transfer across a wide variety of problems, to be creative, and tackle tasks never seen during training. They are far from displaying human-like capabilities in terms of open-ended learning. This is, perhaps, because they rely on isolated signals for learning. The way forward might be to build on child development theory and to consider learning from *sociocultural interactions*. Indeed humans are social beings, they interact and cooperate with their peers (Tomasello, 1999b; Tomasello et al., 2005; Brewer et al., 2014). As soon as they discover and learn a language, they assimilate thousands of years of experience embedded in their culture (Bruner, 1991). Most of their skills could not be learned in isolation. Formal education teaches them to reason systematically, books teach them history, and YouTube might teach them how to cook.

Most importantly, humans' values, traditions, norms, and most of their goals are cultural in essence.

The present research proposes to immerse artificial agents in social contexts in order to observe the impact of sociocultural interactions on learning. As displayed in Fig. 1.1, it has a dual objective. In the first part of this manuscript, we propose to use artificial agents as an anthropological tool to study the formation of cultural conventions in populations of individuals. More specifically, we investigate the key mechanisms required for the self-organization of cultural conventions between artificial agents in absence of pre-existing conventions. In the second part, we focus on autonomous artificial agents exploiting already existing cultural conventions to augment their capabilities in the open-ended skill acquisition problem. To accomplish this, we build on previous theories at the intersection of developmental psychology and machine learning to introduce a new framework coined *Vygotskian Autotelic Artificial Intelligence* which enables sociocultural interactions to transform agents' learning signal, yielding better learners.

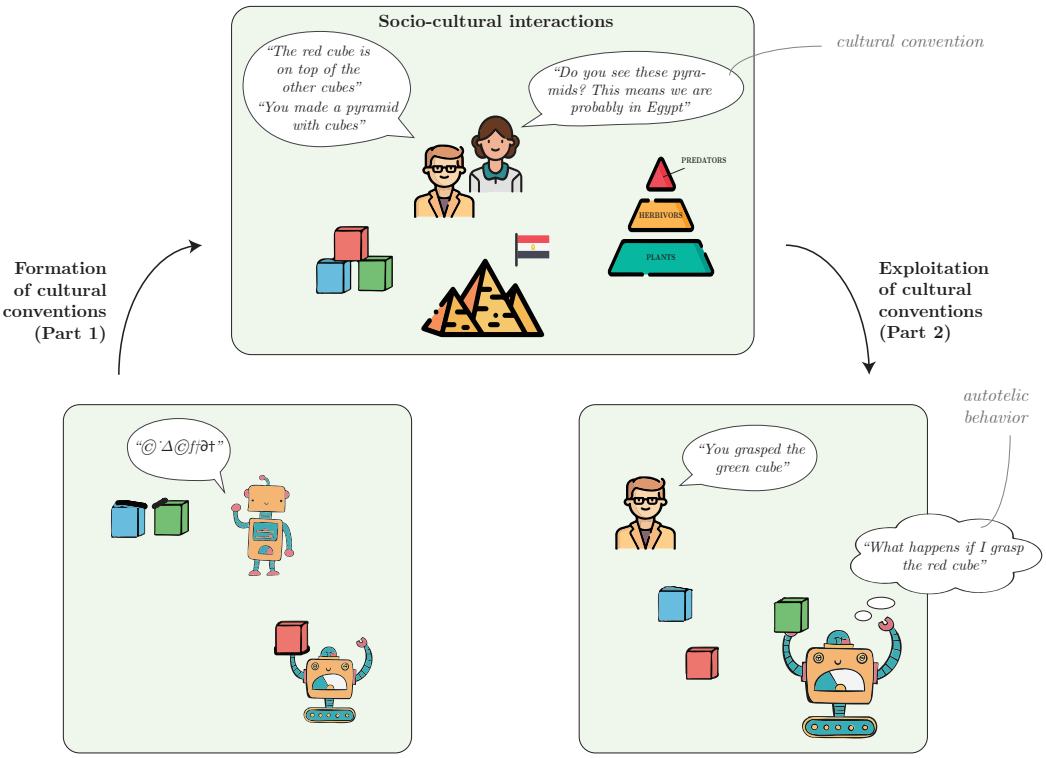


Figure 1.1: **Dual organization of the present research.** In the first part we take a bottom-up approach and study the self-organization of cultural conventions in artificial agents from social interactions. In the second part, we use a top-down approach to investigate the impact of pre-existing cultural conventions on artificial agents when they interact with social peers.

The remaining of this introduction presents key features of human learning that enable us to define the important notions of "autotelic learning" and "cultural convention" at the center of this research. We then close it with a short intuitive explanation of the position of this research with respect to other paradigms in AI and a summary of our contributions.

1.1 Humans are goal-directed social learners

Humans are an incredible source of inspiration for AI. They are the fastest learning system we can ever witness. Within only a few years, children learn to crawl and navigate their home, identify and manipulate objects, they even learn to speak and interact with their peers. How do they reach such a level of proficiency in such a short period of time?

1.1.1 Humans are autotelic learners

A central aspect of human development is the notion of goal. Studying the use of the notion of goal in past psychological research, Elliot & Fryer (2008) propose the following general definition:

> "*A goal is a cognitive representation of a future object that the organism is committed to approach or avoid* " (Elliot & Fryer, 2008).

A goal is therefore a future projection that influences human behaviors. During exploratory play, children constantly invent and pursue their own problems/goals (Chu & Schulz, 2020). In particular, children's exploration seems to be driven by intrinsically motivated brain processes that trigger spontaneous exploration for the mere purpose of visiting interesting situations (Gopnik et al., 1999; Kaplan & Oudeyer, 2007; Kidd & Hayden, 2015b). But how do we measure interestingness? Hunt (1965) propose to evaluate situations in term of *optimal incongruity*. Similarly, Berlyne (1966) suggest relying on the notion of *intermediate level of novelty* while Kidd et al. (2012) showed that young infants focus on goals with *intermediate complexity*. Finally, Csikzentmihalyi (1997), in his flow theory suggests that for human beings to feel pleasure during learning they should target goals with *optimal challenge*. He uses the term *autotelic* to describe intrinsically motivated agents that are in the flow state.

Definition

Autotelic: from the Greek *auto* (self) and *telos* (end, goal), characterizes agents that generate their own goals and learning signals. It is equivalent to *intrinsically motivated and goal-conditioned*.

1.1.2 Humans are social learners

Social interactions are another crucial property of human development. At birth, humans enter a culture that strongly shapes their development (Whorf, 1956). Humans are social beings; intrinsically motivated to interact and cooperate with their peers (Tomasello, 1999b; Tomasello et al., 2005; Brewer et al., 2014). Indeed, we use social interactions and language at every stage of our development to communicate, cooperate, teach and organize our thoughts.

Cooperation

First, social interactions enable us to **cooperate**, to jointly commit to shared goals. Tomasello (2019) describes this collaborative behavior as *shared intentionality*. According to him, shared intentionality arises around nine months and enables us to relate to others as equals and to align on low-level common goals such as "looking in the same direction". Shared intentionality allows us to mentally represent and then adopt another's goal. It is thus very linked to the theory of mind (Wellman, 1992). It allows us to share goals, emotions, attention, or even knowledge. As we grow older, shared intentionality becomes *collective intentionality* and allows us to be part of a society in which goals are associated with social norms and conventions. In a recent study, Mcclung et al. (2017) use an egg hunt game to show that group membership and the ability to talk led to increased collaboration between participants. By analyzing the conversation they found that in-group participants were talking about the hunt in terms of a shared or common goal, while out-group participants used individual goals.

Teaching

In a more structured way, social interactions also enable us to **teach**. The idea that social interactions provide a structure for teaching has been supported by many researchers including Vygotsky (1933); Bruner (1985); Rohlfing et al. (2016); Vollmer et al. (2016). Bruner (1985) specifically proposed the concept of *pragmatic frames*: patterns of behaviors that are used to achieve a goal and that are developed through repeated and sequential interactions between a teacher and a learner. According to Bruner, pragmatic frames are made of two key components: 1) a *syntax* which is the observable part of the interactions and includes the sensory means (modalities) as well as the role of each actor; 2) a *meaning* which is the learning content. In his book, Bruner (1985) takes the example of the *book-reading frame* during which the teacher points and asks for labels before providing feedback and correcting the learner depending on their answer. In this case, the pointing/asking/answering mechanism is the syntax and the label is the meaning. Pragmatic frames can happen in a variety of modalities but as we just saw with the book-reading frame, they are often multi-modal and imply linguistic interactions.

Pragmatic frames may also adapt to the learners' abilities. In Vygotsky's *zone of proximal development* (Vygotsky, 1934), caretakers naturally scaffold the learning experiences of children, tailoring them to their current objectives and capacities. Through encouragement, attention guidance, explanations, or plan suggestions, they provide cognitive aids to children in the form of interpersonal social processes. In this zone, children can benefit from these social interactions to achieve

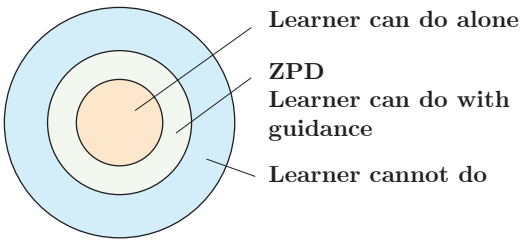


Figure 1.2: ZPD Illustration

more than they could alone as illustrated in Fig. 1.2. In Vygotsky's terms, the ZPD is defined as:

> "*the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem-solving under adult guidance, or in collaboration with more capable peers*" (Vygotsky, 1934)

Thoughts

The language we use in social interaction can also be a cognitive tool that facilitates **thinking**. In Vygotsky's theory, children *internalize* linguistic and social aids and progressively turn these interpersonal processes into intrapersonal *psychological tools*. This essentially consists in building internal models of social partners such that learners can self-generate contextual guidance in the absence of an external one. Social speech is internalized into private speech (an outer speech of children for themselves), which, as it develops, becomes more goal-oriented and provides cognitive aids of the type caretakers would provide (Vygotsky, 1934; Berk, 1994). Progressively, it becomes more efficient and abbreviated, less vocalized, until it is entirely internalized by the child and becomes *inner speech*. This inner speech would enable *thinking in language* (Carruthers, 1998). The relation between language and thought in humans is the subject of a great debate and will be discussed in greater detailed in chapter 6, Sec. 6.2 when introducing the Vyogtskian Auotelic AI framework.

Cultural ratchet

Finally, language is a cultural artefact inherited from previous generations and shared with others. It supports our cultural evolution and allows humans to efficiently transfer knowledge and practices across people and generations (Henrich & McElreath, 2003; Morgan et al., 2015; Chopra et al., 2019) — a process known as the *cultural rachet* (Tomasello, 1999b). Through shared cultural artefacts such as narratives, we learn to share common values, customs and social norms, we learn how to navigate the world, what to attend to, how to think, and what to expect from others (Bruner, 1990).

Cultural Convention

In light of the various properties of social interactions presented in this section, we introduce the notion of *cultural convention* which generalizes pragmatic frames to internal (intrapersonal) social production. More specifically, we propose the following definition.

Definition

Cultural Convention: A social production, linguistic or physical, internal or interpersonal, used to communicate, cooperate, teach, think, or transmit.

Cultural conventions are patterns of behavior that emerge among population of agents to solve repeated coordination problems (Freire et al., 2020). But any interaction used for coordination should not be deemed as cultural conventions. Indeed cultural conventions possess two distinctive properties: (i) self-sustainability, wherein a group of individuals within a given population persist in adhering to a specific convention as long as they anticipate others to follow it, and (ii) arbitrariness, whereby alternative and equally plausible resolutions exist to address the identical problem.

1.2 Towards Interactive Social Autonomous Agents

The present research aims at bridging developmental psychology with recent AI methods used to design embodied artificial agents. Building on the autotelic and cultural convention notions, our goal is to build interactive social autotelic agents. For this purpose, we immerse artificial agents in social contexts and equip them with learning mechanisms to either construct cultural conventions (in part I) or to exploit cultural conventions to discover new skills (in part II).

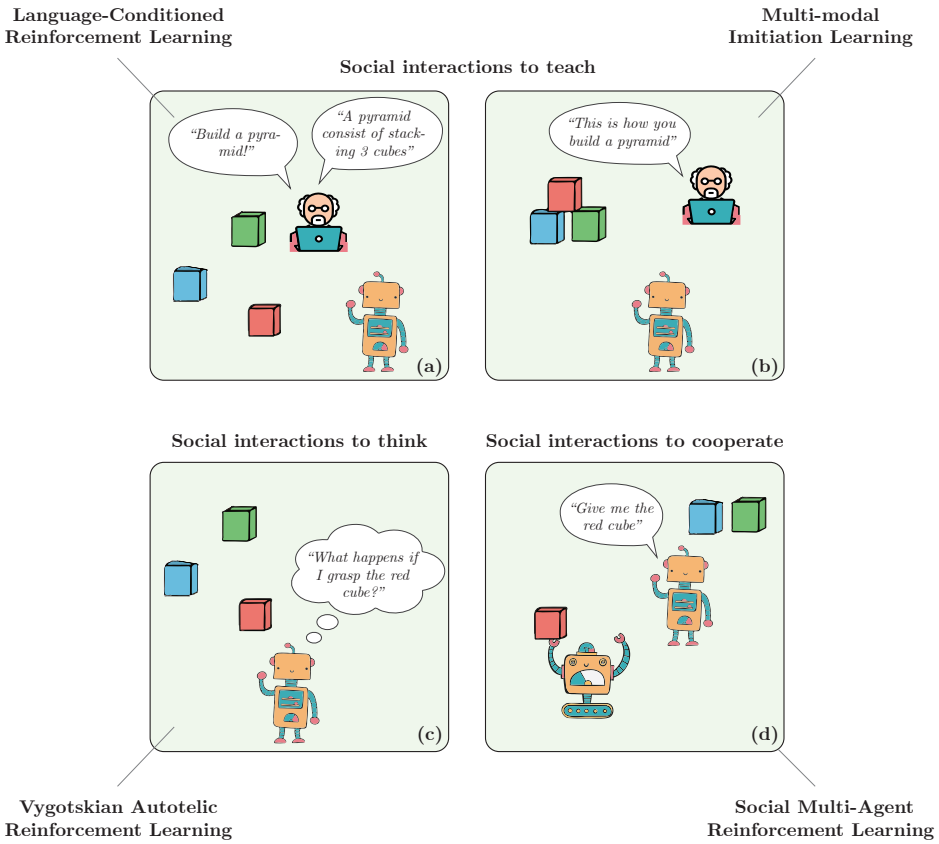


Figure 1.3: **Social interactions in different AI paradigms.** Social interactions and language instructions are used in both RL and IL setting to guide learners. Language can also serve as a cognitive tool to represent goals in autotelic learning. Finally, they can help agents communicate and cooperate in MARL.

The immersion of artificial agents in social worlds does not require starting from

fresh grounds. In fact, numerous works already include social elements in pre-existing AI paradigms. In a recent survey, Luketina et al. (2019) review several approaches instructing RL agents with language, either to condition them or to assist them as displayed in Fig. 1.3 (a). Similarly, recent IL settings have had their training datasets augmented with linguistic descriptions of expert trajectories (Shridhar et al., 2020; Pashevich et al., 2021) as displayed in Fig. 1.3 (b). In the present research, we will demonstrate that agents can use language as a cognitive tool to imagine creative goals (Colas et al., 2022a) as illustrated in Fig. 1.3 (c). Finally, Jaques et al. (2019) recently presented a MARL framework where agents use social motivations to solve collaborative tasks such as the one depicted in Fig. 1.3(d).

Objectives

The objective of the present research is to investigate the two following questions:

- **How can cultural conventions self-organize when artificial agents interact?** The objective of the first part of this research is to investigate the key mechanisms required for the **formation** of cultural conventions between artificial agents. In part I, we place ourselves in a multi-agent setup and consider social interactions between two artificial agents that both integrate learning dynamics.
- **How can artificial agents benefit from pre-existing cultural conventions?** Conversely, part II aims at exploring the **exploitation** of pre-existing cultural conventions by autonomous agents. As such, we will consider a single artificial agent interacting with a simulated social partner.

Contributions

The present manuscript starts with an overview of foundational AI paradigms, namely RL, IL, and MARL (chapter 2). Following this, in chapter 3, we present the two primary research questions we address here, which are organized around the theme of self-organization: 1) self-organization of cultural convention, and 2) self-organization of trajectories derived from existing cultural conventions.

Our first experimental contribution (chapter 4) investigates the role of sensorimotor constraints in the formation of a graphical language. For this experiment, we place ourselves in the context of Language Games (Steels, 2001) and consider speaker and listener agents exchanging utterances to refer to visual objects. In our setup, utterances are graphical signs produced by a robotic arm and objects are combinations of MNIST digits. We propose a new multi-modal contrastive learning algorithm to enable agents to self-organize a shared communication system in such a sensorimotor setting.

Our second experimental contribution (chapter 5) studies the collaboration between two artificial agents in the *Architect-Builder Problem*: a new interactive setting in which agents have asymmetrical roles and must cooperate to build structures. More specifically, the architect knows the structure that needs to be assembled but cannot act on the blocks of the environment while the builder does not know the task at hand but can manipulate the objects. Our proposed algorithmic solution builds on the shared intentionality and

pragmatic frame concepts to enable the architect and the builder to agree on a cultural convention enabling them to solve the task.

Our next contribution (chapter 6) introduces the Vyogtksian Autotelic AI framework (VAAI). Inspired by the pioneering work of the developmental psychologist Vygotsky (1934), we draw the contour of a more human-like AI where agents are immersed in rich socio-cultural worlds. By exposing agents to our culture, and enabling them to internalize pre-existing cultural conventions they can use language as a cognitive tool to become better learners.

The VAAI framework is the foundation of two other experimental contributions. Our fourth contribution (chapter 7) explores how embodied artificial agents can align their trajectories with linguistic descriptions provided by a social partner. This alignment is known as the Language Grounding Problem (Glenberg & Kaschak, 2002b; Zwaan & Madden, 2005b). We consider the grounding of descriptions involving spatio-temporal concepts and study the impact of architectural biases by testing different variants of multi-modal transformers.

Finally, in our fifth and last contribution (chapter 8), we implement an autotelic agent that converts linguistic descriptions given by a social partner into targetable goals. We coined this agent IMAGINE. IMAGINE operates in two phases. First, the agent learns to represent, detect and achieve goals by interacting with a social partner. Once it has discovered a variety of interesting interactions doing so, IMAGINE then switches to an autonomous phase and uses language as a cognitive tool to imagine new goal constructs leveraging language compositionality. We show that this algorithm enables agents to discover a greater variety of skills paving the way to more open-ended learning learners.

How to read this manuscript

We propose to organize our contribution in a linear and systemic fashion. We first explore how artificial agents can self-organize cultural conventions from tabula-rasa in part I. Then we assume pre-existing conventions to investigate how they can impact skill acquisition in part II. This linear progression starts with the introduction of the CURVES algorithm as an ecological way to learn graphical cultural conventions and ends with the presentation of the IMAGINE agent that leverages cultural conventions to creatively explore its environment. However, this manuscript can also be read backward. Starting from the observation that cultural conventions are a necessary condition for the design of open-ended learners, understanding the formation of cultural conventions becomes of primordial importance. It can, for instance, help develop new learning scenarios for agents such as the Architect Builder problem (chapter 5), and new inductive biases for learning architectures such as the multi-modal transformers (chapter 7). As a matter of fact, the contributions constituting this research are presented in anti-chronological order. My research journey started with the development of the IMAGINE agent and ended so far with the introduction of CURVES.

Fig. 1.4 illustrates how one can navigate through this manuscript.

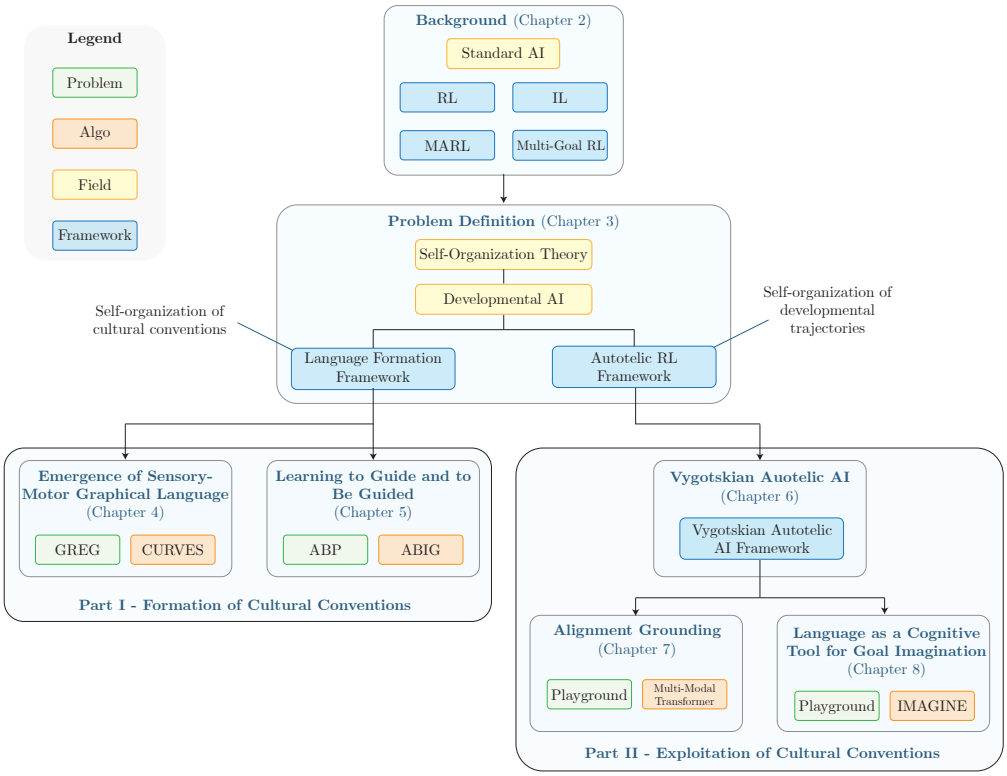


Figure 1.4: **How to read this manuscript?** This manuscript starts by introducing the standard AI frameworks. It then presents the two general problems we investigate belonging to the field of developmental AI. The problems are formulated as the language formation framework and the autotelic RL framework. The two parts of the manuscript extend each of these frameworks. Note that the term framework has an ambivalent definition. It refers simultaneously to families of problems and their associated solutions. The specific problems we investigate are illustrated with green boxes while our algorithmics contributions are in orange boxes.

1.2.1 Collaborations

The present research is the result of multiple collaborations involving several research institutions including INRIA in France, Mila in Canada, and Microsoft Research at Cambridge (UK). My two amazing supervisors, Clément Moulin-Frier and Pierre-Yves Oudeyer from the Flowers Lab (INRIA) were involved in all these collaborations. Our first contribution (chapter 4) was developed during the brilliant internship of Yoann Lemesle (Paris-Dauphine-PSL University) which I had the chance to supervise. Our second contribution (chapter 5) was led by the great Paul Barde (Mila) and myself, under the joint supervision of my and Paul Barde's supervisors, namely Derek Nowrouzezahrai (McGill University) and Chris Pal (Polytechnique Montreal & Mila, CIFAR AI Chair). Most of the work on Vygotskian Autotelic Agents presented in chapter 6 and 8 was conducted in close collaboration with Cédric Colas (INRIA) who acted as a mentor at the beginning of my thesis, providing me all the tools to carry out efficient research. More specifically, the IMAGINE approach was developed in collaboration with Nicolas Lair (INSERM, Cloud Temple), Peter-Ford Dominey (INSERM), and Jean-Michel Dussoux (Cloud Temple). Finally, our work on grounding spatio-temporal language with transformers (chapter 7)

is the result of a project with Laetitia Teodorescu (INRIA) and her supervisor Katja Hofman (Microsoft Research).

1.2.2 Publications

Journals

- Autotelic Agents with Intrinsically Motivated Goal-Conditioned Reinforcement Learning: A Short Survey, *Journal of Artificial Intelligence Research* 74 (2022), 1159-1199. Colas et al. (2022b) (Co-author)
- Language and Culture Internalisation for Human-Like Autotelic AI, *Nature Machine Intelligence* (2022) Colas et al. (2022a) (Co-first-author)

Conferences

- Language as a Cognitive Tool to Imagine Goals in Curiosity-Driven Exploration, *Advances in Neural Information Processing Systems* 33 (2020). Colas et al. (2020a) (Co-first-author)
- Grounding Spatio-Temporal Language with Transformers, *Advances in Neural Information Processing Systems* 34 (2021). Karch et al. (2021) (Co-first-author)
- Learning to Guide and to Be Guided in the Architect-Builder Problem, *International Conference on Learning Representations* (2022). Barde et al. (2022) (Co-first-author)

Workshops

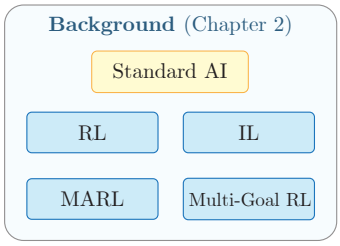
- Deep Sets for Generalization in RL, *ICLR 2020 workshop Beyond tabula rasa in reinforcement learning: agents that remember, adapt, and generalize.* Karch et al. (2020) (Co-first-author)
- Language-Goal Imagination to Foster Creative Exploration in Deep RL, *ICML 2020 workshop Language in Reinforcement Learning.*

Pre-print

- Contrastive Multimodal Learning for Emergence of Graphical Sensory-motor Communication (2023). Karch et al. (2023) (Co-first-author)

Chapter 2

Background: Standard AI Paradigms



Contents

Our contributions bridge standard AI paradigms and developmental psychology to investigate two fundamental research questions (1) the language acquisition problem (self-organisation of cultural conventions) and (2) the open-ended skill acquisition problem (self-organisation of trajectories). In this chapter, we review the standard AI problems and their associated families of algorithmic solutions.

2.1 Reinforcement Learning

Problem

In a Reinforcement Learning problem, an agent learns to perform sequences of actions in an environment by maximizing some notion of cumulative reward (Sutton & Barto, 2018). The agent interacts with the environment in the form of a temporal sequence unfolding from time $t = 0$ to time $t = T$, T being the episode horizon and representing the lifetime of the agent (potentially variable or infinite). RL problems are commonly framed as Markov Decision Processes (MDPs).

Definition

Markov Decision Process (MDP):

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0, R\} \tag{2.1}$$

where $\mathcal{S}$ and $\mathcal{A}$ are respectively the state and action spaces, $\mathcal{T}$ is the transition function that dictates how actions impact the world (lead to the next state), ρ_0 is the initial state distribution and R is the reward function.

At the beginning of an episode, the agent starts in the initial state $s_0 \sim \rho_0(\mathcal{S})$. At each time step the agents takes action $a_t \in \mathcal{A}$ and observes the next state $s' = s_{t+1} \in \mathcal{S}$ and the reward $r_{t+1} = R(s_t, a_t)$. A diagram of interaction is given in Fig. 2.1. The

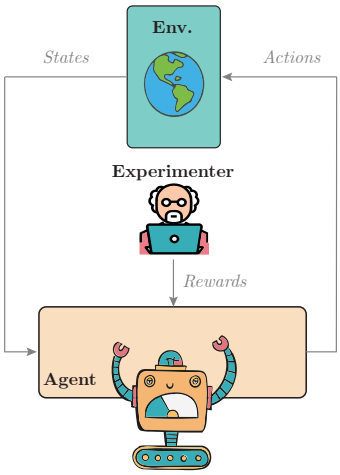


Figure 2.1: Interactions in a RL loop

transition function $\mathcal{T}$ gives the distribution of the following states from the current state and action: $\mathcal{T} = P_E(.|s, a)$ with P_E being the (potentially stochastic) dynamics of the environment. In an MDP, the transition function must respect the *Markov property*: a future state (s') must only depend on the current state (s) and not on its predecessor, i.e. the transition function is memoryless.

$$P_E(s_{t+1}|s_t, a_t) = P_E(s_{t+1}|s_0, \ldots, s_t, a_t) \tag{2.2}$$

In a RL problem, the behavior of the agent is expressed as a *policy* $\pi : \mathcal{S} \to \mathcal{A}$ that predicts the next action a based on the current state s. This policy can be stochastic $a_t \sim \pi(.|s_t)$ or deterministic $a_t = \bar{\pi}(s_t)$. When agents interact in an environment, they produce *trajectories*. A trajectory is a sequence of states and actions $\tau = (s_0, a_0, \ldots, s_T, a_T)$. When both the dynamics of the environment and the policy of the agent is stochastic, the probability of a trajectory is:

$$P(\tau|\pi) = \rho_0(s_0) \prod_{t=0}^{T-1} P_E(s_{t+1}|s_t, a_t)\pi(a_t|s_t) \tag{2.3}$$

The objective of the agent is to maximize the cumulative reward computed over trajectories (R^{tot}). When computing the aggregation of rewards, we often introduce discounting

and give smaller weights to delayed rewards. The return of a trajectory is therefore:

$$R^{tot}(\tau) = \sum_{t=0}^{T} \gamma^t R(s_t, a_t) \tag{2.4}$$

with $\gamma \in]0, 1]$ being a constant discount factor. We call the optimal policy π^*, the behavior that maximizes the expected return:

$$\pi^* = \underset{\pi}{\text{argmax}} \underset{\tau \sim \pi}{\mathbb{E}} [R^{tot}(\tau)] = \underset{\pi}{\text{argmax}} \underset{(a_t \sim \pi, s_t \sim P_E)}{\mathbb{E}} \left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right] \tag{2.5}$$

The reward function plays therefore a crucial role in a RL problem as its maximization will directly shape the behavior of the agent.

Value Functions

Most RL algorithms rely on the definition of *value* and *action-value* functions:

Definitions

- The Value Function $V_\pi(s)$ of a policy π gives the expected return of a trajectory starting from s and following π.
- The Action-value Function $Q_\pi(s, a)$ is the expected return of the trajectory taking action a from state s before following π from the next state s'.

Action-value functions are powerful because they allow us to instantly assess the quality of a situation without waiting for the end of the trajectory. The value and action-value function obey the Bellman expectation equations (Sutton et al., 1998), a recursive definition that states that the value of a certain state (when following policy π) is equal to the sum of the instantaneous reward and the value from the next state.

$$\begin{cases} V_\pi(s) = \underset{(a \sim \pi, s' \sim P_E)}{\mathbb{E}} [R(s, a) + \gamma V_\pi(s')] \\ Q_\pi(s, a) = \underset{s' \sim P_E}{\mathbb{E}} \left[R(s, a) + \gamma \underset{a' \sim \pi}{\mathbb{E}} [Q_\pi(s', a')] \right] \end{cases} \tag{2.6}$$

The value and action-value functions also follow the Bellman optimality equation where expectations over actions are replaced by max operators.

$$\begin{cases} V^*(s) = \max_a \underset{s' \sim P_E}{\mathbb{E}} [R(s, a) + \gamma V^*(s')] \\ Q^*(s, a) = \underset{s' \sim P_E}{\mathbb{E}} \left[R(s, a) + \gamma \max_{a'} [Q^*(s', a')] \right] \end{cases} \tag{2.7}$$

Acting greedily with respect to the optimal action-value function gives the optimal policy:

$$\pi^*(s) = \underset{a}{\text{argmax}}\, Q^*(s, a) \tag{2.8}$$

Computing Q^* is therefore a way to solve a RL problem. When agents have access to perfect knowledge of the dynamic of the environment (P_E) and when the dimensionality of $\mathcal{S}$ and $\mathcal{A}$ is small, they can do planning to find the optimal action-value function via

Dynamic Programming (Bellman, 1966) for instance. Planning approaches that leverage the transition function of the environments are called *model-based* RL algorithms. They are opposed to *model-free* RL algorithms that do not use P_E but interact directly with a simulator (with transition function P_S).

Because the present research builds on both families of solutions, we detail the techniques used for each in the following paragraphs. We first briefly detail the *Monte-Carlo Tree Search* planning algorithm (MCTS) (Browne et al., 2012) used in our first experimental contribution (in chapter 5) and then introduce the deep RL algorithm used in chapter 8.

Model-based RL with MCTS:

MCTS is a tree-search algorithm that seeks to identify the optimal policy by finding the action with the highest Q-value. To this end, MCTS builds an estimate $\hat{Q}(s,a)$ for $a \in \mathcal{A}$ in a given state s and acts greedily with respect to this estimate. Each node of the tree is a state s while edges are the potential actions. The MCTS algorithm grows the tree iteratively using an exploration/exploitation tradeoff to efficiently refine $\hat{Q}$ in promising regions of the MDP. More specifically, each iteration of the MCTS algorithm contains four steps:

1. **Selection:** In the selection phase, the MCTS algorithm starts from the root node and uses a tree policy to decide which node to expand. The tree policy is guided by an evaluation function (UCT) and stops when a node with remaining actions to explore is reached.
2. **Expansion:** Once a leaf node is reached, a new action a is sampled among the non-explored ones and the corresponding node is computed using the transition function $s' \sim P_E(.|s,a)$
3. **Simulation:** From the newly created node corresponding to state s', a simulation policy π_{sim} is used to draw a full trajectory (until termination or for a predefined horizon) and compute return R^{tot}. π_{sim} is often a random policy.
4. **Backpropagation:** R^{tot} is backpropagated to the root node as indicated in Fig. 2.2.

For the tree policy evaluation function, we use the Upper Confidence Bound (Auer et al., 2002): $UCT = \frac{1}{k}\sum_{i=0}^{k} R_i^{tot} + C\sqrt{\frac{\ln(n)}{k}}$ where k is the number of completed trajectory going through node s and n is the number of iterations. The first term of UCT is an estimation of the expected return while the second term encourages the tree policy to explore unexpanded nodes.

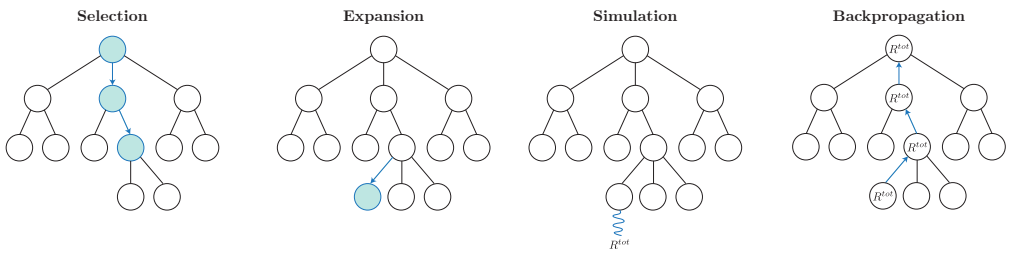


Figure 2.2: The four steps of an MCTS iteration

Model-free RL with Q-learning:

Some of the experimental contributions of this research build on the *Deep Deterministic Policy Gradient* (DDPG) algorithm (Lillicrap et al., 2016). DDPG derives from *Deep Q-Networks* (DQN) (Mnih et al., 2015) which is itself a deep learning implementation of the standard *Q-learning* algorithm (Watkins & Dayan, 1992). In this paragraph, we propose to detail the steps that allow building DDPG from Q-learning.

Q-learning is an *off-policy* RL algorithm. Off-policy algorithms, in contrast to on-policy algorithms, learn to approximate the action-value Q^* of an optimal policy independently of the policy used for data collection. Q-learning relies on transitions (s, a, r, s') collected by a policy π_c interacting with a simulator P_S. Assuming that Q is a linear combination of features (ϕ): $Q(s, a; \theta) = \theta^T \phi(s, a)$, the algorithm iteratively learns to approximate Q^* by minimizing the temporal difference error (TD-error):

$$\mathcal{L}_i = \mathop{\mathbb{E}}_{(s\sim P_S, a\sim\pi_c)} \left[(y_i - Q(s, a; \theta_i))^2\right] \text{ with } y_i = \mathop{\mathbb{E}}_{s'\sim P_S} \left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1})\right] \tag{2.9}$$

In the original formulation of the Q-learning algorithm by Watkins & Dayan (1992), they consider a tabular setting and store the Q-values at each iteration in a table ($Q_i[s, a]$) instead of using linear function approximations. The update of the table writes:

$$Q_{i+1}[s, a] \leftarrow Q_i[s, a] + \alpha \left(r + \gamma \max_{a'} Q_i[s', a'] - Q_i[s, a]\right) \tag{2.10}$$

DQN proposes to represent the action-value function with deep neural networks: $Q(s, a; \theta)$ with parameters θ. The architecture of the network takes a state s as input and outputs the value of each action $Q(s, a) \forall a \in \mathcal{A}$. Thus DQN only works with discrete action space. When differentiating Eq. (2.9) with respect to the neural network parameters, we get:

$$\nabla_{\theta_i} \mathcal{L}_i(\theta_i) = \mathop{\mathbb{E}}_{(s\sim P_S, a\sim\pi_c)} \left[(y_i - Q(s, a; \theta_i)) \nabla_{\theta_i} Q(s, a; \theta_i)\right] \tag{2.11}$$

During differentiation, one has to pay particular attention to freezing the weights of the network when evaluating y_i. Deep neural networks are known to exhibit training instabilities. In order to stabilize learning, Mnih et al. (2015) proposed two main innovations:

- *Experience Replay*: The agent uses a replay buffer to store transitions during interactions. During learning, the transitions are then sampled uniformly to perform updates. This enables breaking the correlation between successive transitions and reusing them.
- *Target network*: A target network is used to compute target y. This network is initialized with the actual Q-network ($Q_{targ}(s, a; \theta_{targ}) = Q(s, a; \theta)$ but updated less frequently than the actual Q-network. Updates are often performed using *Polyak averaging* (Polyak & Juditsky, 1992): $\theta_{targ} \leftarrow \rho\theta_{targ} + (1 - \rho)\theta$ with ρ being the polyak factor.

DDPG is an adaptation of DQN to continuous action space. The challenge of dealing with continuous actions is to act greedily with respect to the learned Q-value. i.e. to evaluate $\text{argmax}_a Q(s, a)$. To overcome this, DDPG concurrently learns a deterministic

policy with the Q-function. This policy is a parametrized network $\pi(s;\phi)$ with parameters ϕ and is obtained by gradient ascent. Moreover, since $\pi(s;\phi) \approx \text{argmax}_a\, Q(s,a,\theta)$ it can be injected in Eq. (2.9). We, therefore, have the two following losses to optimize:

$$\begin{cases} \mathcal{L}_{\pi_\phi} = \mathop{\mathbb{E}}_{(s\sim P_S)} \left[Q_\theta(s, \pi_\phi(s))\right] & \text{(Policy loss)} \\ \mathcal{L}_{Q_\theta} = \mathop{\mathbb{E}}_{(s\sim P_S, a\sim\pi_c)} \left[(y - Q_\theta(s,a))^2\right] & \text{(Q-value loss)} \\ \quad \text{with } y = \mathop{\mathbb{E}}_{s'\sim P_S} \left[r + \gamma Q_\theta(s', \pi_\phi)\right] \end{cases} \tag{2.12}$$

where parameter dependencies have been subscripted.

Other model-free RL algorithms

There are numerous algorithms within the field of Deep RL, including on-policy methos like TRPO (Schulman et al., 2015), PPO (Schulman et al., 2017) as well as more advanced off-policy approaches like TD3 (Fujimoto et al., 2018) and SAC (Haarnoja et al., 2018).

2.2 Imitation Learning

Problem

Imitation Learning (IL) (Pomerleau, 1988; Schaal, 1996; Osa et al., 2018) is a field that considers an agent learning in a MDP in which the reward function is not explicitly defined, but where the agent can observe demonstrations of the task it is intended to perform. IL is particularly useful in situations where it is difficult for the experimenter to design a task-specific reward function, but demonstrations are available. A classic example from the literature is the application of IL to self-driving cars. It is impractical to specify a reward function for the task of driving as successful drivers constantly adjust their criteria to adapt to the various events that occur on the road. However, there is a vast amount of video footage of people driving that could potentially be utilized by the agent to learn. A diagram of interactions of the IL problem is provided in Fig. 2.3

A standard way of formalizing the IL problem is to find a policy that minimizes the divergence between the expert and learner data distribution. Provided a dataset $\mathcal{D} = \{(\tau_i)\}_{i=1}^N$ containing expert trajectories of features $\tau = [\phi_0, \ldots, \phi_T]$. If $q_{\pi^*}(\phi)$ is the distribution of features induced by the expert's policy (supposed optimal π^*) and $p_\pi(\phi)$ is the distribution of features induced by the learners' policy (π), the goal of IL is to find policy $\hat{\pi}$ such that:

$$\hat{\pi} = \mathop{\text{argmin}}_{\pi} D(q_{\pi^*}(\phi), p_\pi(\phi)) \tag{2.13}$$

with D being a measure of differences between probability distributions such as the well-known Kullback-Leibler (KL) divergence.

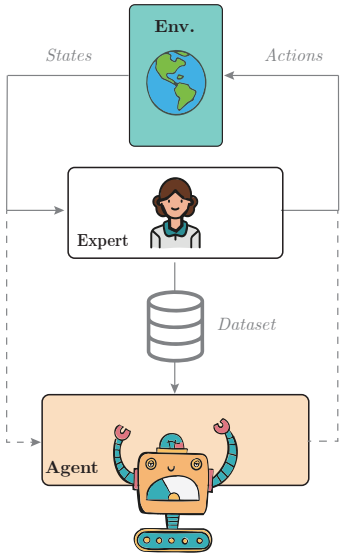


Figure 2.3: Interactions in a IL problem. The agent never interacts with the environment during learning but can interact with it to test its behavior (dashed lines).

Behavioral Cloning

An intuitive way of solving an IL problem is to frame it as a supervised learning setting and do *Behavioral Cloning* (BC). Given a dataset of trajectories $\mathcal{D} = \{(\tau_i)\}_{i=1}^{N}$ with $\tau = [(s_0, a_0) \dots (s_T, a_T)]$, one directly minimizes the cross entropy loss:

$$\mathcal{L}_\pi = - \mathop{\mathbb{E}}_{(s,a) \sim \mathcal{D}} [\log \pi(s, a)] \tag{2.14}$$

Minimizing this cross-entropy loss is in fact equivalent to minimizing the KL-divergence between the trajectory distribution of the expert $P(\tau|\pi^*)$ and the trajectory distribution of the learner $P(\tau|\pi)$ (Ke et al., 2020):

$$D_{KL}(P(\tau|\pi^*), P(\tau|\pi)) = \sum_{\tau \in \mathcal{D}} P(\tau|\pi^*) \log \left(\frac{P(\tau|\pi^*)}{P(\tau|\pi)} \right) \tag{2.15}$$

Injecting the definition of the trajectory distribution of Eq. (2.3) we get that:

$$D_{KL}(P(\tau|\pi^*), P(\tau|\pi)) = \sum_{\tau \in \mathcal{D}} P(\tau|\pi^*) \log \left(\prod_{t=0}^{T-1} \frac{\pi^*(a_t|s_t)}{\pi(a_t|s_t)} \right) \tag{2.16}$$

$$= \sum_{\tau \in \mathcal{D}} P(\tau|\pi^*) \sum_{t=0}^{T-1} (\log \pi^*(a_t|s_t) - \log \pi(a_t|s_t)) \tag{2.17}$$

$$= \mathop{\mathbb{E}}_{(s,a) \sim \mathcal{D}} [\log \pi^*(a_t|s_t) - \log \pi(a_t|s_t)] \tag{2.18}$$

We will use behavioral cloning in chapter 5. BC is a straightforward method for reproducing expert behavior. However, simple BC only works if the agent operates in the same region of the state space as the states provided in $\mathcal{D}$. Otherwise, the policy of the learner will progressively deviate from this region accumulating errors at each time step. This compounding error is called *distributional mismatch*. One way of addressing it is to

iteratively collect new expert data when needed (in the initially uncovered region of the state space) (Ross et al., 2011).

Another limitation of BC is that it is only able to derive an optimal policy from optimal expert trajectories, meaning that the learned policy will not exceed the performance of the expert. In some applications collecting optimal trajectories is not always possible. As a result, some researchers have turned to *Inverse Reinforcement Learning* (IRL) as an alternative approach.

Inverse Reinforcement Learning

Similar to RL, IRL can be understood both as a problem and a category of techniques. The IRL problem consists in recovering the reward function of an expert given a dataset of its trajectories (Ng & Russell, 2000). As such IRL algorithmic solutions followed by RL can form a solution to the IL problem. The combination of IRL followed by RL is called *Apprenticeship Learning* (Abbeel & Ng, 2004). As opposed to BC, apprenticeship learning ensures that the learned policy is bellman consistent (with respect to an underlying learned value function). As formalized by Klein et al. (2011), there are mainly three categories of strategies to obtain the policy in apprenticeship learning:

1. Feature-expectation-based methods as proposed by Ziebart et al. (2008) which learn a reward function such that the feature expectation of the optimal policy (according to the learned reward function) is similar to the feature expectation of the expert policy.
2. Margin-maximization-based methods (Ratliff et al., 2006), which formulate IRL as a constrained optimization problem in which the expert's examples have a higher expected cumulative reward than all other policies by a certain margin.
3. Approaches based on the parameterization of the policy by the reward (Neu & Szepesvári, 2007): If it is assumed that the expert follows a Gibbs policy (or the optimal value function related to the optimized reward function), it is possible to estimate the likelihood of a set of state-action pairs provided by the expert.

Recent feature-expectation-based approaches use technics similar to generative adversarial networks (GAN) (Goodfellow et al., 2014) to imitate complex behavior in high-dimensional environments (Ho & Ermon, 2016). Other approaches use a ranking of trajectories to reach better-than-demonstrator performances (Brown et al., 2020a). As we do not leverage IRL in our contributions we will not detail these methods (see Arora & Doshi (2021) for a thorough survey of IRL algorithms).

2.3 Muli-Goal Reinforcement Learning

Standard RL can be extended to a multi-goal setting. Let us return to the definition of goal by Elliot & Fryer (2008) provided in the introduction (Sec. 1.1.1):

> "*A goal is a cognitive representation of a future object that the organism is committed to approach or avoid*".

RL algorithms seem, indeed, to be a good fit to train goal-conditioned agents: they train learning agents (*organisms*) to maximize (*approach*) a cumulative (*future*) reward (*object*). In Multi-Goal RL, goals can be seen as a set of *constraints* on one or several consecutive states that the agent seeks to respect. These constraints can be very strict and characterize a single target point in the state space (e.g. image-based goals) or a specific sub-space of the state space (e.g. target x-y coordinate in a maze, target block positions in manipulation tasks). They can also be more general when expressed by language for example (e.g. '*find a red object or a wooden one*').

Formal Definition of Goals and Skills

To represent these goals, Multi-Goal RL agents must be able to 1) have a compact representation of them and 2) assess their progress towards it. This is why we propose the following formalization for goals:

Generalized definition of the goal construct for Multi-Goal RL:

- Goal: a $g = (z_g, R_g)$ pair where z_g is a compact *goal parameterization* or *goal embedding* and R_g is a *goal-achievement* function.
- Goal-achievement function: $R_g(\cdot) = R_{\mathcal{G}}(\cdot \mid z_g)$ where $R_{\mathcal{G}}$ is a goal-conditioned reward function.

The objective of a goal-conditioned agent is to learn a *goal-conditioned policy*: a function that generates the next action given the current state and the goal $a_t \sim \pi(\cdot|s_t, z_g)$. The goal-achievement function and the goal-conditioned policy both assign *meaning* to a goal. The former defines what it means to achieve the goal, it describes how the world looks like when it is achieved. The latter characterizes the process by which this goal can be achieved; what the agent needs to do to achieve it. In this search for the meaning of a goal, the goal embedding can be seen as the map: the agent follows this map and via the two functions above, experiences the meaning of the goal.

Definition

- Goal-conditioned policy: a function that generates the next action given the current state and the goal.
- Skill: the association of a goal and a policy to reach it.

Problem

By replacing the unique reward function R by the space of reward functions $\mathcal{R}_{\mathcal{G}}$ in the definition of MDP of Eq. (2.1), RL problems can be extended to handle multiple goals: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0, \mathcal{R}_{\mathcal{G}}\}$. The term *goal* should not be mistaken for the term *task*, which refers to a particular MDP instance. As a result, *multi-task* RL refers to RL algorithms that tackle a set of MDPs that can differ by any of their components (e.g. $\mathcal{T}, R, \rho_0$, etc.). The *multi-goal* RL problem can thus be seen as the particular case of the multi-task

RL problem where MDPs differ by their reward functions. In the standard multi-goal RL problem, the set of goals — and thus the set of reward functions — is pre-defined by engineers. As one can observe in Fig. 2.4, the experimenter sets goals to the agent, and provides the associated reward functions.

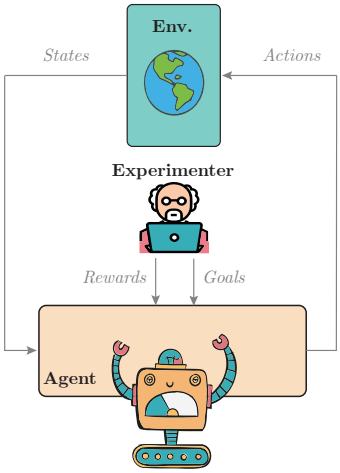


Figure 2.4: Interactions in a multi-goal RL loop. The experimenter provides goals and their associated rewards to the agent.

Solutions: Horde, UVFA, and HER

Goal-conditioned agents see their behavior affected by the goal they pursue. This is formalized via goal-conditioned policies, that is policies that produce actions based on the environment state and the agent's current goal:

$$\Pi : \mathcal{S} \times \mathcal{Z}_{\mathcal{G}} \to \mathcal{A} \tag{2.19}$$

where $\mathcal{Z}_{\mathcal{G}}$ is the space of goal embeddings corresponding to the goal space $\mathcal{G}$ (Schaul et al., 2015). Note that ensembles of policies can also be formalized this way, via a meta-policy Π that retrieves the particular policy from a one-hot goal embedding z_g (Kaelbling, 1993; Sutton et al., 2011).

The idea of using a unique RL agent to target multiple goals dates back to (Kaelbling, 1993). Later, the HORDE architecture proposed to use interaction experience to update one value function per goal, effectively transferring to all goals the knowledge acquired while aiming at a particular one Sutton et al. (2011). In these approaches, one policy is trained for each of the goals and the data collected by one can be used to train others.

Building on these early results, Schaul et al. (2015) introduced *Universal Value Function Approximators* (UVFA). They proposed to learn a unique goal-conditioned value function and goal-conditioned policy to replace the set of value functions learned in HORDE. Using neural networks as function approximators, they showed that UVFAs enable transfer between goals and demonstrate strong generalization to new goals.

The idea of *hindsight learning* further improves knowledge transfer between goals (Kaelbling, 1993; Andrychowicz et al., 2017a). Learning by hindsight, agents can reinterpret

a past trajectory collected while pursuing a given goal in the light of a new goal. By asking themselves, *what is the goal for which this trajectory is optimal?*, they can use the originally failed trajectory as an informative trajectory to learn about another goal, thus making the most out of every trajectory (Eysenbach et al., 2020). This ability dramatically increases the sample efficiency of goal-conditioned algorithms and is arguably an important driver of the recent interest in goal-conditioned RL approaches.

A typology of Goal Representations

The concept of goal in Multi-Goal RL is a central aspect of the autotelic RL framework that we will detail in Sec. 3.3. Therefore we, here, propose to review the different kinds of goal representations found in the literature. For each category of goal, we detail the form of the goal embedding and the reward function.

Goals as choices between multiple objectives. Goals can be expressed as a list of different objectives the agent can choose from. This is the case in Oh et al. (2017); Mankowitz et al. (2018); Codevilla et al. (2018); Chan et al. (2019b).

Goal Embedding	*Reward Function*
z_g are one-hot encodings of the current objective being pursued among the N objectives available. z_g^i is the i^{th} one-hot vector: $z_g^i = (\mathbb{1}_{j=i})_{j=[1..N]}$.	The goal-conditioned reward function is a collection of N distinct reward functions $R_{\mathcal{G}}(\cdot) = R_i(\cdot)$ if $z_g = z_g^i$.

Goals as target features of states. Goals can be expressed as target features of the state the agent desires to achieve.

Goal Embedding	*Reward Function*
A state representation function φ maps the state space to an embedding space $\mathcal{Z} = \varphi(\mathcal{S})$. Goal embeddings z_g are target points in $\mathcal{Z}$ that the agent should reach.	$R_{\mathcal{G}}$ is based on a distance metric D. The reward can be dense: $R_g = R_{\mathcal{G}}(s\|z_g) = -\alpha \times D(\varphi(s),\ z_g)$, or sparse: $R_{\mathcal{G}}(s\|z_g) = 1$ if $D(\varphi(s),\ z_g) < \epsilon$, 0 otherwise.

In manipulation tasks, z_g can be target block coordinates (Andrychowicz et al., 2017a; Nair et al., 2018a; Plappert et al., 2018; Colas et al., 2019a; Fournier et al., 2021; Blaes et al., 2019; Lanier et al., 2019; Ding et al., 2019; Li et al., 2020). In navigation tasks, z_g can be target agent positions (Schaul et al., 2015; Florensa et al., 2018). Agent can also target image-based goals. In that case, the state representation function φ is usually implemented by a generative model trained on experienced image-based states and goal embeddings can be sampled from the generative model or encoded from real images (Zhu et al., 2017; Codevilla et al., 2018; Nair et al., 2018b; Pong et al., 2020; Warde-Farley et al., 2019; Florensa et al., 2019; Venkattaramanujam et al., 2019; Lynch et al., 2020; Lynch & Sermanet, 2020; Nair et al., 2020; Kovač et al., 2020).

Goals as abstract binary problems. Some goals cannot be expressed as target state features but can be represented by *binary problems*, where each goal is expressed as a set of constraints on the state that are either verified or not.

Goal Embedding	*Reward Function*
z_g can be any expression of the set of constraints that the state should respect. Akakzia et al. (2021a); Ecoffet et al. (2021) propose a pre-defined discrete state representation. Another way to express sets of constraints is via language-based predicates	The reward function of a binary problem can be viewed as a binary classifier that evaluates whether state s (or trajectory τ) verifies the constraints expressed by the goal semantics (positive reward) or not (null reward)

When goals are expressed in language, a sentence describes the constraints expressed by the goal, and the state or trajectory either verifies them or does not (Hermann et al., 2017a; Chan et al., 2019a; Jiang et al., 2019a; Bahdanau et al., 2019a,c; Hill et al., 2020a; Cideron et al., 2020c; Colas et al., 2020b; Lynch & Sermanet, 2020), see Luketina et al. (2019) for a recent review. Language can easily characterize *generic goals* such as "*grow any blue object*" (see chapter 8), *relational goals* like "*sort objects by size*" (Jiang et al., 2019a), "*put the cylinder in the drawer*" (Lynch & Sermanet, 2020) or even *sequential goals* "*Open the yellow door after you open a purple door*" (Chevalier-Boisvert et al., 2019a). When goals can be expressed by language sentences, goal embeddings z_g are usually language embeddings learned jointly with either the policy or the reward function.

Goals as a multi-objective balance. Finally, some goals can be expressed, not as desired regions of the state or trajectory space but as more general objectives that the agent should maximize. In that case, goals can parameterize a particular mixture of multiple objectives that the agent should maximize

Goal Embedding	*Reward Function*
z_g are sets of weights balancing the different objectives $z_g = (\beta_i)_{i=[1..N]}$ where β_i is the weights applied to objective i and N is the number of objectives.	The reward is expressed as a convex combination of objectives: $R_g(s) = \sum_{i=1}^{N} \beta_g^i R^i(s)$ where R^i is the i^{th} of N objectives and $z_g = \beta = \beta_i^g \mid_{i\in[1..N]}$ is the set of weights.

In *Never Give Up*, for example, RL agents are trained to maximize a mixture of extrinsic and intrinsic rewards (Badia et al., 2020b). The agent can select the mixing parameter β that can be viewed as a goal. Building on this approach, AGENT$_{57}$ adds control of the discount factor, effectively controlling the rate at which rewards are discounted as time goes by (Badia et al., 2020a).

2.4 Multi-Agent Reinforcement Learning

Problem

Standard RL can also be extended to scenarios where several agents interact with the environment. For this purpose MDPs are extended to *Markov Games*.

Definition

Markov Game are defined by the following terms:

$$\mathcal{M} = \{\mathcal{S}, \mathcal{T}, \rho_0, \{\mathcal{O}_i, \mathcal{A}_i, R_i\}_{i=1}^{N}\} \tag{2.20}$$

The first three terms of a Markov Game are the same as those of a MDP: $\mathcal{S}$ is the state space, $\mathcal{T}$ is the transition function, and ρ_0 the initial state distribution. However, each agent (denoted by the index i) perceives a different perspective of the state through observation transformation $\mathcal{O}_i$. Agents also have different action spaces $\mathcal{A}_i$ and reward function R_i

In Multi-Agent Reinforcement learning (MARL), each agent aims at learning a policy that maps their observation $o_i = \mathcal{O}_i(s)$ to actions: $a_i \sim \pi_i(\cdot|o_i)$. Similarly to RL, each agent aim at maximizing its expected return:

$$\pi_i^* = \underset{\pi_i}{\text{argmax}} \underset{(a_t \sim \pi_i, s_t \sim P_E}{\mathbb{E}} \left[\sum_{t=0}^{T} \gamma^t R_i(\mathcal{O}_i(s_t), a_t) \right] \tag{2.21}$$

A diagram of interaction is provided in Fig. 2.5. The field of MARL considers mainly two types of tasks:

- *Cooperative tasks* where the agents pursue the same goal and need to coordinate in order to solve it. Cooperative tasks are usually hard to design and often involve the maximization of a common objective (sometimes at the expense of individual gains). For a review of cooperative MARL see OroojlooyJadid & Hajinezhad (2019).
- *Competitive tasks* where the agents pursue non-aligned goals. In these settings agents explicitly aim at maximizing their individual gains.

Among the recent innovations in MARL, Baker et al. (2020) trained agents to play the hide-and-seek game, Pérolat et al. (2017) to solve common-pool resource problems, and more recently Stooke et al. (2021) trained an agent on a spectrum of cooperative and competitive tasks including cooperative games to find objects, hide and seek or even capture the flag.

Solution

One of the main challenges of multi-agent learning systems is to take into account the non-stationary dynamics caused by the change of state of the agents when they learn. Indeed, an isolated agent of a Markov game does not evolve in a stationary MDP because all agents are learning, and their behavior will be different during training. For

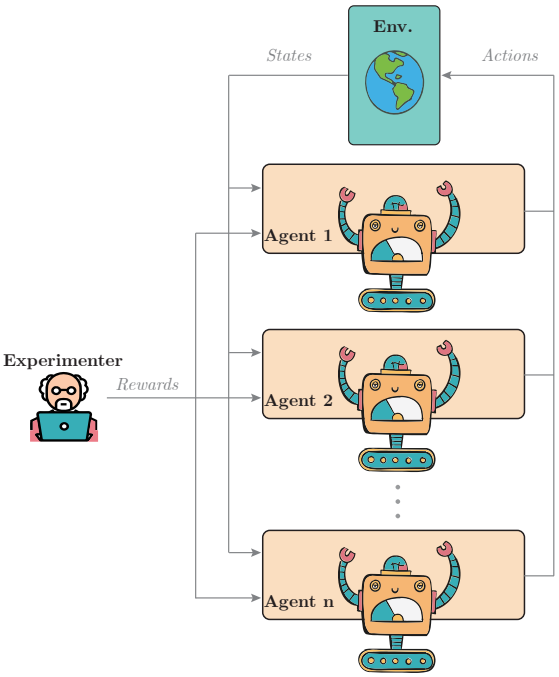


Figure 2.5: Diagram of interactions in a MARL loop. Each agent perceives a (potentially) different perspective of the states provided by the environment. Each agent also has its own action space and is given a (potentially) different reward.

this reason, most of the MARL algorithms rely on the *centralized training, decentralized execution* paradigm. For instance, Muli-Agent Deep Deterministic Policy gradient (Lowe et al., 2017), uses a centralized training procedure where all agents can see other agents' observations and actions to learn an action-value function that is then used to optimize decentralized policies that only depends on local observations. As none of our contributions builds on MARL we will not elaborate on other MARL algorithms.

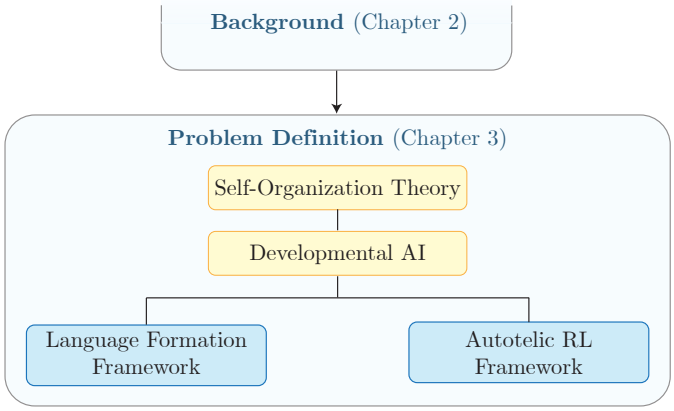
2.5 Summary

In this chapter, we presented four standard AI paradigms and provides a toolbox of techniques and concepts, which can, hopefully, serve as a technical reference for readers who are not familiar with certain notions while reading this manuscript. The next chapter will extend upon these paradigms to investigate the self-organization of cultural conventions between artificial agents and the role of cultural conventions in the self-organization of agents' developmental trajectories.

More specifically, the RL framework will serve to build computational models of language formation (presented in 3.2.1 of chapter 3). The RL and IL frameworks will be leveraged to investigate the emergence of goal-directed communication in the architect-builder problem presented in chapter 5. The multi-goal RL framework is extended in chapter 3 Sec. 3.3 to build the autotelic RL framework which is itself at the origin of the Vyogtskian autotelic RL presented in chapter 6.

Chapter 3

Problem Definition: Developmental AI



Contents

The present research expands upon the standard AI methods presented in the previous chapter, with the aim of investigating fundamental inquiries within the domain of Developmental Artificial Intelligence. Developmental AI is a multidisciplinary field that integrates principles from artificial intelligence, developmental psychology, linguistics,

and neuroscience to simulate and analyze the cognitive development of sensorimotor, cognitive, and cultural structures, both at the level of artificial agents and at the level of populations. While standard AI paradigms are structured around precise and formal problems addressed by algorithmic contributions, developmental AI strives to create machine systems that can learn in an autonomous, autotelic, and open-ended manner, similar to the way children learn. In this research the specific questions that we investigate are: 1) How do cultural conventions emerge through interactions among agents in social contexts? 2) How can autotelic artificial agents utilize cultural conventions to acquire open-ended skill repertoires? These questions can be approached through the lens of self-organization theory. Specifically, this study will examine: 1) the self-organization of conventions, or language, among agents, and 2) the role of cultural conventions in the self-organization of agents' developmental trajectories.

The initial part of this section (Sec. 3.1) outlines the fundamental concepts of open-ended learning, self-organization, and compositionality which are central to this research. After having presented these notions and their relations to our two scientific questions, this chapter act as a bifurcation and formally poses the problems we will tackle in the two separate parts of this manuscript. Sec. 3.2 presents a typology of the language formation framework which provides us a structured approach to categorizing our two first computational studies (presented in Sec. 3.2.2), namely the self-organization of graphical sensory-motor language, and the formation of cultural convention in the Architect-Builder problem. These contributions will be developed in Part I of this manuscript. Then, Sec. 3.3 describes the open-ended formation of skill repertoire and introduces the autotelic RL framework. The autotelic RL framework will serve as a basis to explore the role of cultural conventions in the self-organization of developmental trajectories. We outline our conceptual and computation contributions on this topic in Sec. 3.3.2 and detail them in Part II of this manuscript.

3.1 Open-ended Learning, Self-organizing Systems, and Compositionality

The purpose of this section is to provide a formal definition of the three key concepts, namely open-ended learning, self-organization, and compositionality. Prior to defining these concepts, this section aims to present a global overview of why they are significant and how they relate to one another within this manuscript.

To provide insight into why the present research considers the three notions discussed, it is essential to focus on the long-term and meta-objective of developmental AI, which involves designing open-ended learners. This objective raises two fundamental questions: 1) How do open-ended learning processes self-organize, both in terms of individual development and during socio-cultural interactions? and 2) What is the role of compositionality in open-ended learning, both as an emergent property of self-organizing communication systems and as a cognitive tool for learning? Therefore, this section first provides an explanation of open-ended learning before delving into the concepts of self-organization and compositionality.

3.1.1 Open-ended Learning

Open-ended learning refers to a developmental kind of learning where the objectives are not predetermined, but rather the learner is encouraged to discover knowledge and skills through an open exploration process. Open-ended learning is often defined in opposition to direct learning which targets the systematic acquisition of externally-defined knowledge and skills (Hannafin et al., 1994). In open-ended learning, the learner is not provided with a fixed set of answers, but instead, they are encouraged to engage in self-directed exploration, experimentation, and collaboration to discover new insights and knowledge. This type of learning is particularly useful in promoting lifelong learning and adaptability in a rapidly changing world, as it allows learners to develop skills that are not specific to any particular domain but can be applied in various contexts. Recent work in AI investigate how agents immersed in vast worlds (with highly diverse tasks and varying topology) can learn a variety of skills (Stooke et al., 2021).

In the second part of this manuscript, we will discuss the role of cultural convention in open-ended learning. Language productivity plays a significant role in open-ended learning exploration by enabling learners to express and communicate their ideas effectively. In an open-ended learning environment, learners are often required to engage in complex and abstract reasoning, which may require them to generate and articulate their ideas in novel ways. Language productivity allows learners to develop and use language to create new concepts, explore different perspectives, and generate multiple possible solutions to problems Vygotsky (1934). This notion will be further explored in the second part of this manuscript both theoretically with the presentation of the Vygotskian autotelic AI framework (in chapter 6) and empirically with the IMAGINE agent in chapter 8.

3.1.2 Self-organization Theory

Self-organization is a term now used in a variety of sciences that can be described with the following definition:

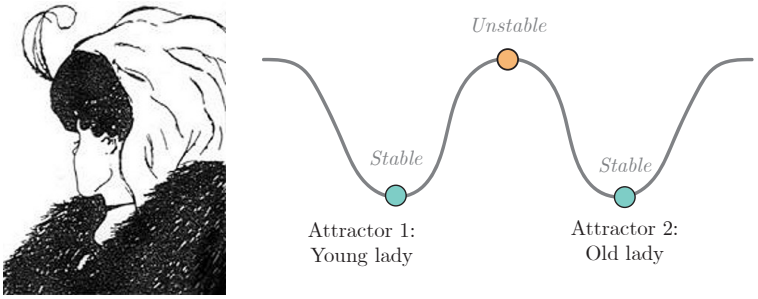
Definition

Self-organization is a process by which spontaneously ordered patterns and structures emerge from the interactions of the many constituents of a system without the need for central control or external guidance. Crucially, the emergent global structure of self-organizing systems has different properties than its local constituents.

Paradoxically, the notion of *emerging order* draws its origin from the study of chaos and was originally used to describe thermodynamical systems that spontaneously organize themself from complex chaotic interactions. The theory of self-organization was formalized by cybernetician Ashby (1962). Borrowing concepts from dynamical system theory, he stated that any complex dynamical systems organize themselves around specific 'attractors' within a vast landscape of possible states. These attractors are stable equilibrium points and may be multiple for a given system. An intuitive explanation of attractors, proposed by Dilts (1995), is given in Fig. 3.1. It illustrates how our complex perception system can fall into different attractors when presented with an illusion. In one case, we see a young woman wearing a necklace looking up to the left, in the other

we perceive an old woman leaning slightly forward. This illustration also demonstrates that certain attractors may be difficult to reach. Indeed, when looking at Fig. 3.1 we often rapidly converge to one attractor and have difficulty escaping it to organize our perception around the other. Finding an attractor requires exploring the landscape of possible states. For this, noise and stochasticity can help as described by von Foerster (2003):

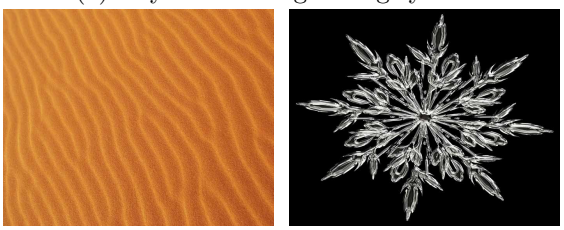
> "*I think it is favorable to have some noise in the system. If a system is going to freeze into a particular state, it is inadaptable and this final state may be altogether wrong. It will be incapable of adjusting itself to something that is a more appropriate situation.*"



(a) (b)

Figure 3.1: (a) My Wife and My Mother-In-Law, by the cartoonist W. E. Hill, 1915 (b) Stability plots illustrating the two attractors of the cartoon as proposed by Dilts (1995)

(a) Physical self-organizing systems



(b) Biological self-organizing systems

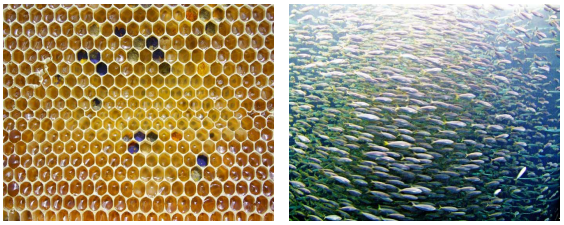


Figure 3.2: Example of self-organizing systems (a) sand dunes in Namibia and crystal structure of snow ice; and (b) a bee comp and a fish school. Images are royalty-free and obtained from pixabay.com

As displayed in Fig. 3.2, nature is full of examples of self-organizing systems. Physical systems exhibit self-organizing behavior through the formation of patterns, such as the longitudinal stripes of sand dunes or the crystalline structure of snowflakes. Similarly, biological systems display self-organizing behaviors(Camazine et al., 2001), as exhibited in the social organization of fish schools and insect swarms, as well as in their ability to collectively adapt to and modify their environment when termites construct mounds and bees build their hives.

Self-organization also enables creative technical innovation such as the development of self-organizing traffic lights (Ferreira et al., 2010): lights that can adapt to changing traffic conditions through local interactions, rather than relying on communication or external signals. Self-organization is particularly well suited for the problem of traffic light regulation because traffic conditions change constantly. Thus, the problem at hand requires adaptation, a property well captured by self-organization theory.

Self-organization in Developmental AI

Developmental AI problems can be formulated as adaptive problems where one or more agents and the environment are coupled dynamical systems whose interactions are responsible for the agents' behavior (Beer, 1995). In this research, we propose to use the language of dynamical systems and the theory of self-organization to formalize the two fundamental problems of this research.

First, the problem of the emergence of cultural convention among artificial agents can be analyzed as the self-organization of a language community (Steels, 1995b; Oudeyer, 2005). A cultural convention is thus an attractor of a language community: when multiple agents interact, variations of language behaviors are attracted to an equilibrium state because the more members of a community adopt a particular convention, the stronger the convention becomes. We will present several approaches that model language formation in Sec. 3.2.

Second, the problem of autonomous skill acquisition can be framed as the self-organization of agents' trajectories where agents use internal mechanisms to develop rather than being controlled by hierarchical top-down control (Pfeifer et al., 2007). In this context, agents develop and grow repertoires of skills via internal drivers and physical interactions with their environment. These internal drivers, referred to as intrinsic motivations (Oudeyer, 2005), allow agents to self-organize their behavior into developmental trajectories and enable them to acquire increasingly complex skills. We will explore the open-ended formation of skill repertoires and present the autotelic approach as a solution to it in Sec. 3.3. We will build on autotelic RL to propose a new vision in developmental RL where agents also leverage social interactions to augment their autonomous learning capabilities

3.1.3 Compositionality

Both parts of this manuscript deal with artifical agents being exposed to cultural conventions in the form of symbolic/linguistic stimuli with the particularity that in part I,

the meaning associated with the conventions are emergent while in part II they are predefined. Semantics (Montague, 1970) is a field at the intersection of linguistics, philosophy, and computer sciences that studies the formation of meanings and their associations with language. A central component of semantics is the concept of Compositionality: our ability to understand language and create meaningful expressions by assembling other meaningful expressions.

Compositionality in Semantics

The topic of our capacity to engage in compositional thinking, as well as the presence of compositional structure in the process of meaning formation is a subject of debate among researchers in the field of semantics. In an entry in the Stanford Encyclopedia of Philosophy, Szabó (2022) provide an overview of the various positions taken by philosophers on this issue and present three arguments in support of the concept of compositionality.

Productivity. The first argument is *productivity* and goes back to Frege's principle: our ability to **concatenate** known meanings to form new ones. In Frege's words:

> "*The possibility of our understanding sentences which we have never heard before rests evidently on this, that we can construct the sense of a sentence out of parts that correspond to words.*" (Frege, 1980)

Thanks to language productivity the potential number of utterances in any human language is infinite. This is illustrated by the famous sentence from Chomsky (1957b): "*Colorless green ideas sleep furiously*". This idea that meaning can be formed in an unbounded manner is related to the notion of open-ended learning, described in the first paragraph.

Systematicity. The second argument for compositionality is *systematicity* or *systematic generalization*. The intuition behind systematicity is that we can form new meanings by **swapping** pieces of information between constructs we already know. From the two sentences "*The cat is asleep*" and "*The dog is awake*", we can understand the sentence "*The cat is awake*". Systematicity implies learning rules of pattern and applying them to new situations. According to Fodor & Pylyshyn (1988b).: "*The ability to entertain a given thought implies the ability to entertain thoughts with semantically related contents*". A more detail definition is provided by Cummins (1996):

> "*A system is said to exhibit systematicity if, whenever it can process a sentence s, it can process systematic variants of s, where 'systematic variation' is understood in terms of permuting constituents or (more strongly) substituting constituents of the same grammatical category.*(Cummins, 1996)

Methodology. The last argument provided by Szabó (2022) supporting compositionality is our ability to communicate in real-time which implies the use of a light

computational model for the production of meanings. Along the same lines, Pagin & Westerståhl (2010) proposed the *learnability argument*:

> "*It must be possible for a speaker to learn the entire language by learning the meaning of a finite number of expressions, and a finite number of construction forms*"

On the other hand, certain counterarguments contest the notion that the meaning of natural language is formed in a purely compositional manner. Fodor (1988) asserts that the formation of lexical meaning may involve some degree of context-sensitivity, although quantifying such variations proves difficult. For instance, Fodor & Pylyshyn (1988a) uses the sentences "*feed the chicken*" and "*chicken to eat*" to demonstrate that they have distinct meanings due to an "*animal/food ambiguity in 'chicken' rather than a violation of compositionality*".

At the core of the debate over compositionality, and closely connected to the hypothesis that language is not purely compositional, lies the inquiry into the appropriate model of cognitive architecture. On the one hand, symbolic models with compositional components allow for the mathematical modeling of meaning formation. However, such models fall short of capturing idioms, and their rigidity is unsuitable for addressing the noisiness and complexity of natural language. Connectionist models, on the other hand, seem better equipped to handle the noisiness of natural language but their capability to perform compositions is questioned. For more information on the debate on the structure of the cognitive architecture readers can refer to Fodor & Pylyshyn (1988a); Pinker (1988); Smolensky (1990); Chalmers (1993).

Compositionality in Artificial Agents

Recently, AI researchers decided to enter the debate and have begun to examine the extent to which artificial agents, parameterized by neural networks, can learn systematicity by exposure to a compositional language. Notably, recent investigations have focused on systematic generalization studies in navigation tasks framed as a seq-2-seq problem with the SCAN (Lake & Baroni, 2018) and gSCAN(Ruis et al., 2020) benchmarks; visual-question-answering (Bahdanau et al., 2019c) and language-conditioned RL (Hill et al., 2020a).

Some work decided to incorporate symbolic computations into neural models to enhance systematic generalization via inductive biases. This is the case of Neural Module Networks (Andreas et al., 2016) which leverages a pre-trained symbolic parser that converts questions into a composition of neural modules trainable end-to-end to answer questions from images. Extending Neural Module Networks, Mao et al. (2019) propose the *Neuro-symoblic concept learner* and replace the pre-trained parser with a differentiable one allowing to learn symbolic decomposition of questions. However, despite the potential advantages of these neuro-symbolic approaches, thorough empirical studies (Bahdanau et al., 2019c; Ding et al., 2020) have not found them to generalize more effectively than the non-symbolic approaches.

Beyond systematic generalization, Hupkes et al. (2020) put forward a set of tests aimed at establishing a rigorous understanding of the constituent elements of the broad

concept of compositionality. These tests are designed to bridge theories of compositional semantics with contemporary successful neural models of language. Alongside the previously recognized tests of systematicity and productivity, Hupkes et al. (2020) introduce new tests such as substitutability (the capacity to cope with synonyms) and overgeneralization (the ability to apply general rules and identify exceptions) which more closely align with the functioning of natural language. These tests are displayed in Fig. 3.3.

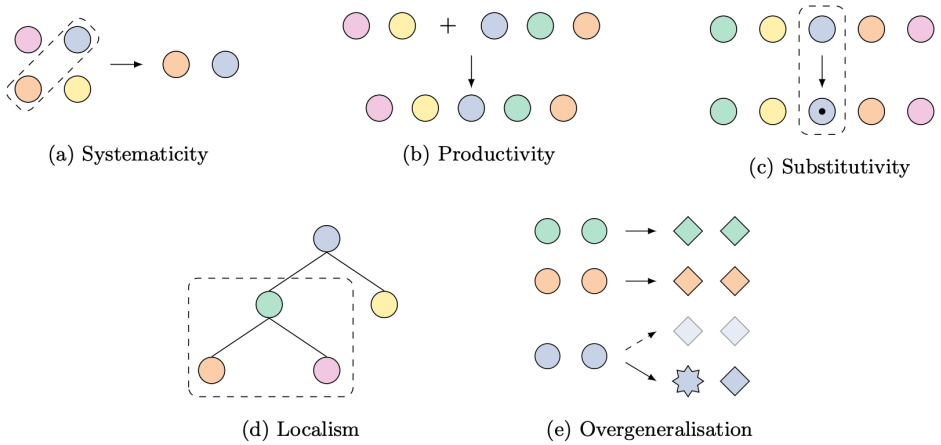


Figure 3.3: **The five compositionality tests as proposed by Hupkes et al. (2020).** (a) the systematicity test evaluates models' ability to recombine known constructs to form new ones; (b) the productivity test evaluates models' ability to concatenate new constructs to form new ones with emphasis on unboundedness; (c) the substitutivity tests agents' capacity to be robust to synonyms; (d) the localism test explores hierarchies in compositions and (e) the overgeneralisation test evaluates how likely models are to infer general rules.

Compositionality in This Research

Compositionality is a proponent topic of this research. In this paragraph, we quickly guide the reader through the different aspects of compositionality that we will discuss in the different parts of this manuscript. In the next section, when presenting the language formation framework, we will review different approaches that tackle the question of compositionality in emergent communication with a particular focus on how the compositional structure of the problem can be reflected in the structure of the emergent language. In our first experimental contribution, in chapter 5, we will perform a productivity test and evaluate a pair of agents on their ability to name compositional referents and explore the relations between productive generalization and compositional structure in the emergent signs. In the second part of this manuscript, we will review numerous works on systematic generalization in language conditioned RL under the light of our proposed Vygotskian autotelic RL framework. In chapter 7 we will investigate the impact of relational inductive biases in the module of an autotelic agent on systematic generalization. Finally, in chapter 8, we will demonstrate how autotelic agents can leverage systematic generalization to explore their environment in an open-ended fashion.

In the preceding paragraph, the concept of compositionality was defined with reference to the domain of semantics. But it should be noted that compositionality is in fact an intrinsic attribute of the world we inhabit. We perceive objects as composites of various attributes (such as size, color, categories, etc.), and our interactions with them are governed by compositional dynamics from which we can extract systematic rules (Plants grow by bringing them water and Cats grow when we bring them food). In our first experimental contribution (chapter 4) we will observe that artificial agents exposed to compositional objects (referents) do not necessarily communicate using a compositional language. Paradoxically, we will demonstrate with our IMAGINE agent (chapter 8) that being exposed to a compositional language is instrumental to fostering exploration via systematic generalization.

3.2 Self-organisation of Cultural Convention: the Language Formation Problem

Now that we have defined the three central concepts of this research, we present a typology of the language formation framework which provides us with a structured approach to categorizing our two computational studies on the formation of cultural conventions between artificial agents.

3.2.1 Computational Models of Language Formation

The study of the origin of language has been a subject of interest and debate among various academic disciplines, including linguistics, archaeology, biology, and anthropology. In this section, we will shortly present the predominant theories on language formation and explore how artificial agents can help experiment with them. For a thorough review of the synthetic modeling of language origins see Steels (1997). There are three predominant theories on the origin of language:

1. The *Genetic evolution theory* postulates that language, just like biological complexity, is the result of natural selection. According to this theory, humans have an innate language organ inside their brains that contains universal rules helping them learn a language during their development. This claim is backed by the famous poverty of stimulus argument which asserts that children do not observe sufficient data to explain their ability to acquire natural language (Chomsky, 1975). The genetic evolution theory thus implies that there exist language genes that code for the language organ and that language is preserved due to genetic transmission.
2. The *Adaptation and self-organization theory* on the other hand supposes that language is preserved in the memories of individuals and transmitted through cultural and social interactions during imitation and acquisition processes. In the adaptation hypothesis, there is no language organ but rather a variety of cognitive and motor primitives that facilitate language formation.
3. The *Genetic assimilation theory* assumes that language is the result of dual dynamics that both involve cultural and genetic interactions. The genetic assimilation hypothesis is also known as the Baldwin effect (Simpson, 1953). It states that learned behaviors that confer a selective advantage can become genetically encoded over time. The genetic assimilation theory proposes that initially, humans did not have an innate language structure and that the first forms of language were acquired through adaptation only. But, if the speed of language acquisition played a role in selection, genetic assimilation would have facilitated the development of language acquisition devices.

Language formation with Artificial Agents

The study of language emergence can benefit greatly from the utilization of agent-based modeling and simulation (Hurford, 1989; Brighton, 2002; Cangelosi & Parisi, 2002; Steels, 2015; Kirby et al., 2014). *Computational Experimental Semiotics* (Galantucci & Garrod, 2011) is a field that analyzes the numerous factors that contribute to language

emergence by examining a population of simulated agents engaging in two distinct types of interaction: *linguistic* and *genetic interactions*. When two agents take part in linguistic interaction, they are in turn speakers and listeners and respectively produce and receive messages describing a context. To study the formation of meanings, linguistic interactions occur within physical environments that contain objects and embodied situations (Steels & Loetzsch, 2012). Depending on the communicative success of linguistic interactions, agents can update their internal state and adapt to their artificial peers. To investigate the impact of population dynamics, the studied population is open: new agents enter, and others leave. These new agents, generated through genetic interactions and subject to potential mutations, introduce an element of novelty into the system. Finally, in order to obtain realistic models, the population should be studied as a distributed multi-agent system, i.e. there should not be any main global agent that acts over the entire population. Moreover, just like humans cannot enter the brain of others, agents should not be able to access each other's internal states. A diagram of interactions as well as a high-level algorithmic implementation of the language formation framework is provided in Fig. 3.4 and Alg. 1.

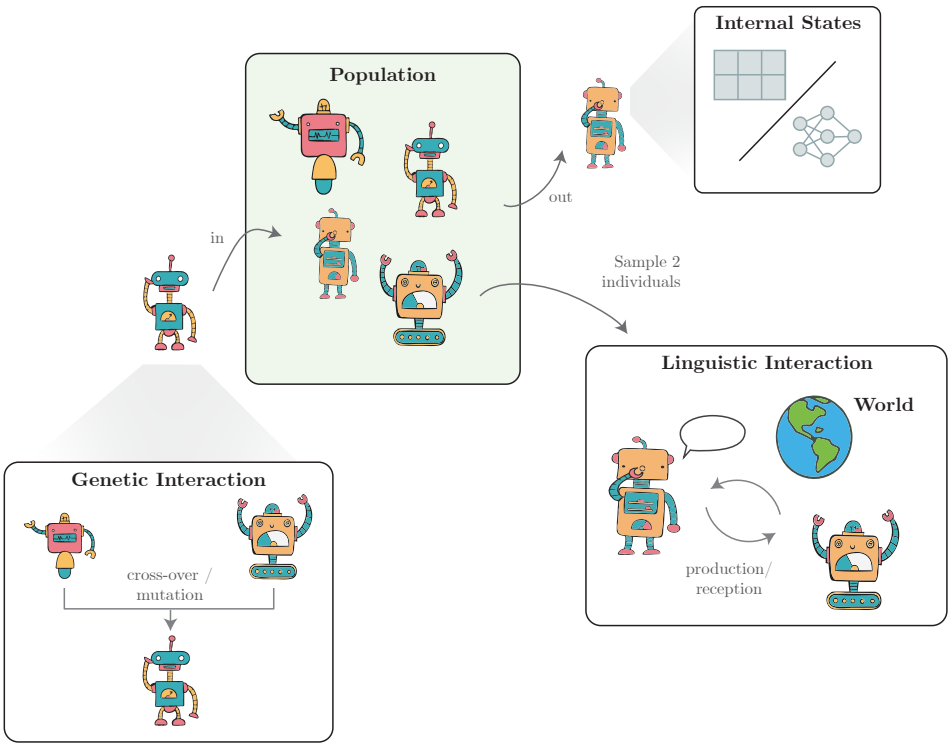


Figure 3.4: **The Language Formation Framework.** A population of agents is an open multi-agent system where new agent enters and others leave. New agents are generated by genetic interactions: crossovers between parents with potential mutations. Agents have internal states allowing them to map signals to actions. They can perform linguistic interactions, i.e. exchanging messages to describe a physical situation of the world.

Algorithm 1: Language formation Simulation

Require: Language Interaction L, Genetic Interaction G, Environment $\mathcal{E}$
  Initialize Population $\mathcal{P}_A$ and internal states of agents
  loop
    Sample two agents from population: $(A_1, A_2) \sim \mathcal{P}_A$
    Store result of linguistic interaction about the world: $\leftarrow L(A_1, A_2, \mathcal{E})$
    Update A_1 and A_2 based on score s
    With prob p_{out}:
      Remove agent from population: $\mathcal{P}_A$.pop()
    With prob p_{in}:
      Sample two parents from population: $(A_1, A_2) \sim \mathcal{P}_A$
      Perform Genetic Interaction: $A' \leftarrow G(A_1, A_2)$
      Add child to population: $\mathcal{P}_A$.add(A')
  end loop

Note that in our contributions, we will not investigate the impact of population dynamics on the emergence of language. Rather, we will focus on the development of agents during a lifetime and disregard any genetic interactions. We will thus focus on the self-organization of cultural conventions during linguistic interactions. We will restrict our analysis to the smallest population of two individuals.

Language Games

The simplest forms of linguistic interaction are coined language games. They derive from *Signaling Games* introduced by Lewis (1969) as a game theoretic approach to the problem of the emergence of conventions. In game theoretic words, a convention is a system of arbitrary rules that enables two players to share meaningful information. Fig. 3.5 presents a simple example of a Lewis game. The two players of a signaling game are the speaker and the listener. In our example, the world is providing two world states to the speaker (w_1 and w_2). Based on the world state, the speaker sends a signal to the listener. Here, there are two available signals (s_1 and s_2). From the received signal, the listener then chose an action (among two actions a_1 and a_2). If the listener picks the correct action for the associated word state then both agents perceive a reward. Note that the Listener never perceives the world state.

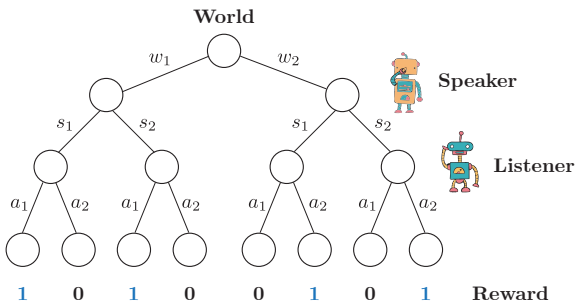


Figure 3.5: Illustration of Lewis Signalling game with two world states, two signs, and two actions.

To investigate the self-organization of conventions around meanings in a more realistic scenario, Steels & Loetzsch (2012) proposed to update signaling games with grounding elements. In a grounded language game, the speaker and the listener are given a shared *context* made of several *referents* (objects) as displayed in Fig. 3.6. The speaker samples a target referent from the context and produces an utterance to name it. Then, the speaker receives the utterance and picks a referent inside the context. If the chosen referent matches the target referent, the game is a success. To self-organize a language, a population of artificial agents needs to play numerous language games. In doing so, agents will alternate between speakers and listers. Depending on the outcome of the game they will update their internal states to reinforce successful conventions and diminish unsuccessful ones. Note that several update strategies are possible. They vary in how the outcome is actually perceived by the agents. On the speaker side, the referent ground truth (target) is known so the outcome of the game can be directly used for the update. On the other hand, since the listener does not know about the target referent some implementations of language games do not communicate the outcome to the listener. In Steels (2001)'s formulation of language game, the outcome is communicated to the speaker via a retroactive pointing mechanism. The speaker basically points toward the target referent at the end of the game to communicate the outcome to the speaker.

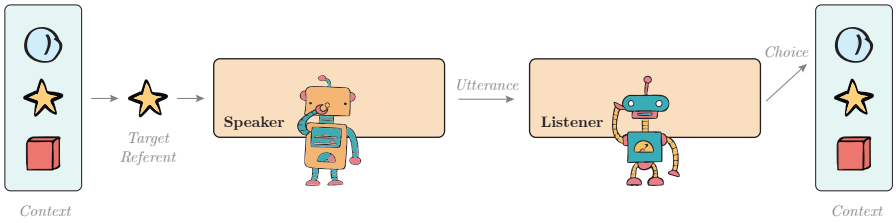


Figure 3.6: Diagram of interactions in a language game

Early solutions to the language game (Steels, 1995b; Oliphant & Batali, 1997; Kirby, 2001) use tables scoring associations between referents and utterances. Given fixed predefined numbers of utterances and referent categories, the agents can adjust the score of utterance/referent association depending on their communicative success. Examples of such tables for the speaker (left) and for the listener (right) are given in Fig. 3.7. If predefined referent categories are not available to the agents, Steels & Loetzsch (2012) propose mechanisms to map visual inputs to object categories. Similarly, the Talking Head experiments (Steels, 2015) propose strategies to adapt the language game to more realistic configurations with flexible and dynamic inventories of words and meanings.

Neural Communicating Agents

Inspired by the success of Convolutional Neural Network in Computer Vision, Lazaridou et al. (2017) proposed to extend language games to image referents with agents using neural networks to take actions [1]. Fig. 3.8 illustrates their setup. The context is made

[1] In deep learning, language games are often referred to as referential games or guessing games

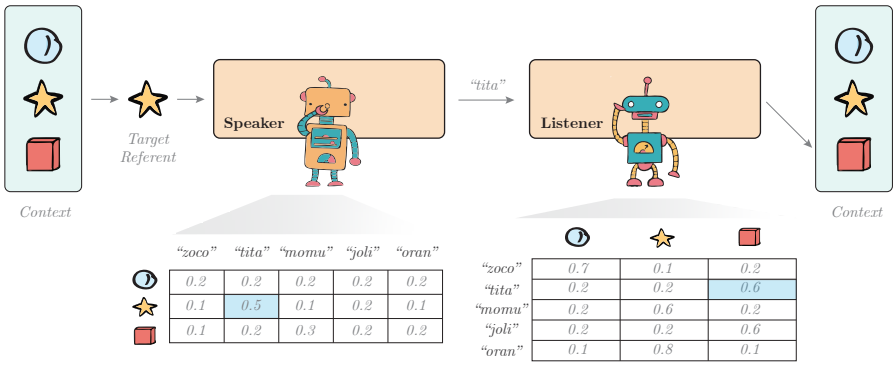


Figure 3.7: Example of agents' tabular internal models, with 3 referents and 5 words.

of two images (i_1 and i_2), a target (t) and a distractor. The utterances are discrete utternaces u coming from a fixed-sized dictionary $\mathcal{V}$. The speaker's utterance is given by a neural network parametrizing a policy that maps the two images to the utterance: $u = \pi_S(i_1, i_2, t; \theta_S)$. Similarly, the listener uses policy π_L to make a choice given the utterance: $a = \pi_L(i_1, i_2, \pi_S(i_1, i_2, t; \theta_S); \theta_L)$. The policies are trained using RL (Sec. 2.1) with reward function R returning 1 iff $\pi_L(i_1, i_2, \pi_S(i_1, i_2, t; \theta_S); \theta_L) := t$. Note that in their implementation the reward and thus the outcome of the game is communicated to both agents which is equivalent to Steel's pointing mechanism.

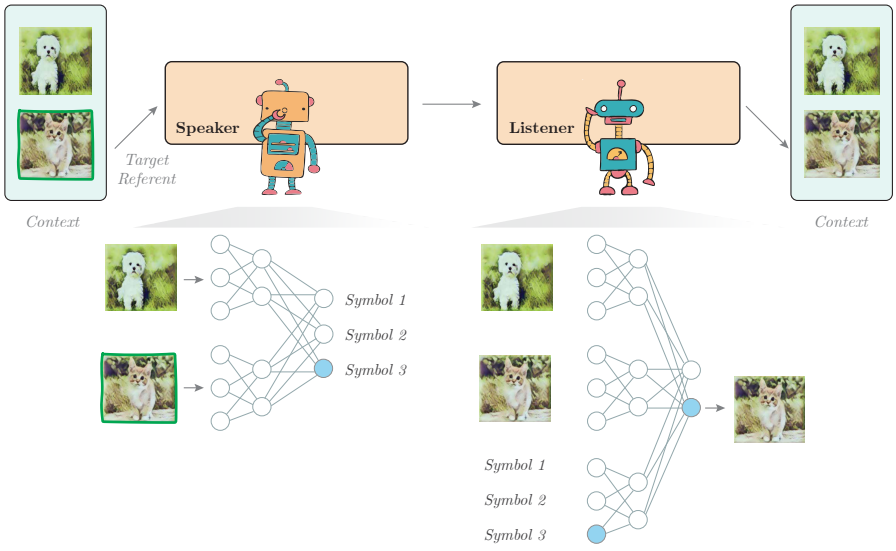


Figure 3.8: Example of agents' neural network internal models, with 2 referents and 3 words (adapted from Lazaridou et al. (2017)).

Beyond scaling previous language game approaches to visual referents, Lazaridou et al. (2017) proposes a strategy to ground the agent's code in natural language. The strategy consists in leveraging a dataset of (image, natural language label) pairs and alternating between RL in the classical language game and standard supervised learning for image classification. In a similar study, Havrylov & Titov (2017) examine the emergence of communication with sequences of symbols and visual referents. They use LSTMs

within the speaker and listener architectures to support the encoding and decoding of discrete tokens arranged in fixed-size sequences. They also analyze the compositionality and variability of the emerging sequences in both tabular-rasa and natural language communication.

The identification of the factors that contribute to the emergence of compositional communication code is a fundamental objective within the field of computational linguistics. To this end, the use of neural communicating agents in language games has emerged as a valuable experimental setting. Kottur et al. (2017) propose to analyze how utterances consisting of sequences of symbols can name referents that are compositions of abstract attributes (represented as one-hot vectors). The decomposition of referents into pre-defined hardcoded attributes enables a more comprehensive and systematic analysis of the compositional properties of the evolving communication code. Building upon this work, Chaabouni et al. (2020) emphasize the importance of separating the compositional generalization capabilities of agents and compositional properties of the emerging code. They have established that the former can be achieved independently of the latter. To complement these systematic analyses, Choi et al. (2018a) look at the emergence of compositional language in a more realistic context where agents perceive different perspectives of the referents. Other works look at the environmental and internal factors that favor the emergence of compositionality. For instance, Rodríguez Luna et al. (2020) show that auxiliary objectives incentivizing object consistency or least effort (the generation of short sequences) support the emergence of compositional code in language games. Similarly, Mu & Goodman (2021) demonstrate that agents solving a variation of the language games where referents are organized in sets of objects agree on a more interpretable and systematic communication code. Finally, Ren et al. (2020) proposed to study the emergence of compositional language in a more complete setting with a population of agents playing language games over several generations.

Goal-Directed Communicating agents

The prior paragraph demonstrates that guessing interactions provide an effective experimental testbed to study language formation. But, as outlined in the introduction, human language serves a multitude of purposes beyond mere object guessing. Therefore, AI researchers have aimed to examine the development of communication in more realistic scenarios, where agents must communicate to accomplish a collaborative task in complex environments that involve interactions with the physical world across multiple time steps. These problems are modeled using MARL as described in Sec. 2.4. The agents must concurrently learn to interact with the world and communicate with others by observing rewards related to their collaborative goal, provided by an expert. See Fig. 3.9 for a visual representation of these interactions. Seminal works on MARL involving communicating agents consider problems such as efficient car coordination at traffic junctions to avoid collision (Sukhbaatar et al., 2016) or riddles where agents need to combine environmental inputs with information communicated over several time steps to succed (Foerster et al., 2016). In their work, Foerster et al. (2016) introduce two approaches for learning to communicate in MARL: Differentiable Inter-Agent Learning (DIAL) and Reinforced Inter-Agent Learning (RIAL). DIAL is based on the centralized training and decentralized execution method and enables gradient to be exchanged between agents, thereby

breaking the assumption of the language formation framework that agents should not be able to have access to each other's internal states. Conversely, in RIAL, messages are viewed as actions produced by a RL algorithm where each agent treats others as a part of the environment, without the need to have access to other agents' internal parameters or to back-propagate gradients. The RIAL algorithm now serves as a baseline for a variety of MARL communication investigations. Jiang & Lu (2018) extended it with an attention mechanism that enables agents to learn when communication is required to solve collaborative tasks. Similarly, Eccles et al. (2019) showed that adding positive signaling (messages must be different in different situations) and positive listening (actions must be different when messages are different) biases to agents via auxiliary losses yields an increase in communicative performance. For a complete survey of emergent communication in MARL setups, see Zhu et al. (2022).

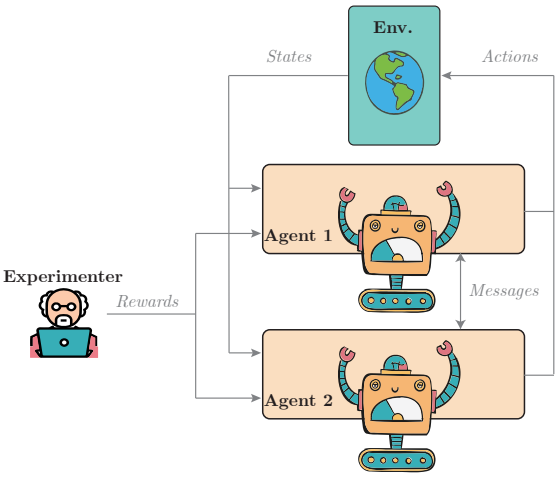


Figure 3.9: Diagram of interactions in MARL emergence communication.

Summary

In this section, we presented the language formation framework which study the emergence of communication inside a population of agents interacting within linguistic and genetic interactions. Our contributions focus on linguistic interactions and ignore the influence of population dynamics on the emergence of communication. We, therefore, presented the most broadly studied linguistic interactions: the language game. Additionally, we showed that this particular guessing interaction setup could be scaled to neural agents. Finally, we noted that MARL offers a valuable framework to study the emergence of communication as a tool to achieve collaborative behaviors in physically complex environments.

3.2.2 Problem Definition

It this now time to turn to our contributions and to formally pose the specific inquiries we target within the context of artificial communicating agents.

Emergence of Graphical Sensory-motor Communication

Our first contribution, that we will present in chapter 4, extends the neural communicating agent framework to consider communication in visual language games via a sensory-motor channel. As reviewed in the previous section, prior approaches focused on agents communicating via an idealized communication channel, where utterances (made of a single or a sequence of symbols) are produced by a speaker and directly perceived by a listener. This comes in contrast with human communication, which instead relies on a *sensory-motor channel*, where motor commands produced by the speaker (e.g. vocal or gestural articulators) result in sensory effects perceived by the listener (e.g. audio or visual). Motivated by this observation we investigate whether artificial agents can develop a shared language in an ecological setting where communication relies on such sensory-motor constraints. To this end, we introduce the *Graphical Referential Game* (GREG) where a speaker must produce a graphical utterance to name a visual referent object while a listener has to select the corresponding object among distractor referents, given the delivered message. See Fig. 3.10 for a diagram of interactions between agents. The utterances are drawing images produced using dynamical motor primitives combined with a sketching library. The referents are images of MNIST (LeCun et al., 1998) digits randomly positioned in the image.

Using sensory-motor systems to examine the development of language dates back to the investigation of the origins of digital vocalization systems in the early 2000s de Boer (2000); Oudeyer (2005); Zuidema & De Boer (2009). However such studies were not conducted in grounded language games. They employed imitation games focusing on the observation of the formation of speech utterances, such as syllables and words, through the systematic combination of lower-level meaningless elements (phonemes). In our study, we chose to focus on a drawing system because 1) conversely to models of vocalization, there is a large number of tools available to researchers to implement realistic sketching mechanisms and 2) it has the advantage of producing 2D trajectories interpretable by humans while preserving the non-linear properties of speech models, which were shown to ease the discretization of the produced signals (Stevens, 1989; Moulin-Frier et al., 2015).

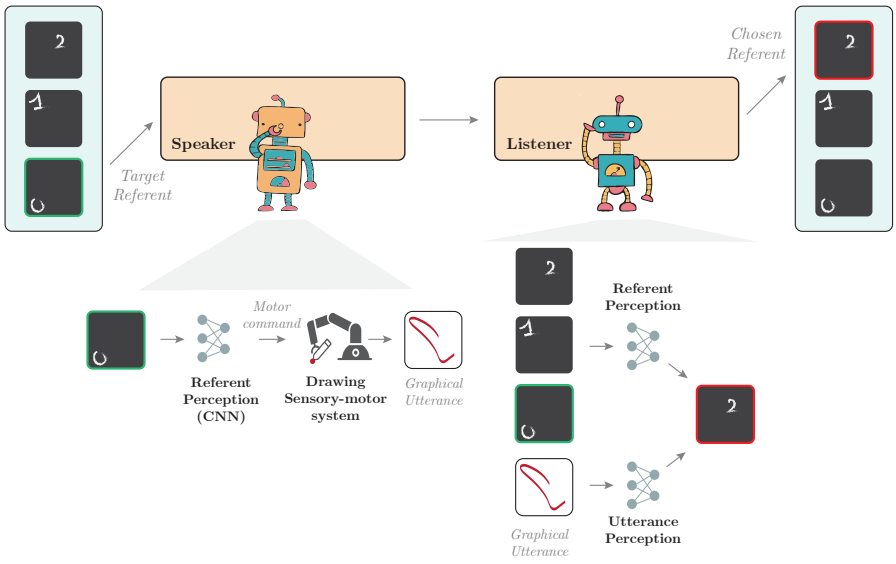


Figure 3.10: The Graphical Referential Game

Studying the GREG we aim at investigating whether a pair of agents can self-organize a shared lexicon from the continuous non-linear constraints of the sensory-motor system. We then propose to study the structure of the emerging signals. We use topographic measures based on a geometric distance to quantify the coherence of the emerging lexicon. Informed by Chaabouni et al. (2020)'s study on the non-equivalence between compositional performance and compositional language, we propose to study these two questions separately. We first evaluate the communicative generalization performances of our system on referents that are the composition of MNIST digits. This is equivalent to performing a productivity test as illustrated in figure 3.3: the agents are trained on 1-digit referents and tested on 2-digits referents. Then we investigate the compositional structure of emerging signs using the same geometry measure as for the coherence.

The Architect-Builder Problem

Our second contribution, that we will develop in chapter 5, proposes to study the *Architect-Builder problem* (ABP), a new AI paradigm that studies the goal-directed emergence of communication in a setup where the reward function is not accessible to all agents. The ABP involves two agents, referred to as the *Architect* and the *Builder*, who must collaborate to accomplish a task. Both agents observe the environment state but only the architect knows the goal at hand. The architect possesses knowledge of the goal and is able to receive the reward associated with it, but is unable to take actions in the environment. In contrast, the builder has no knowledge of the goal or reward and is the only agent that can take actions in the environment. In this asymmetrical setup, the architect can only interact with the builder through a communication signal (messages).

The introduction of the ABP aims to address a gap in the existing literature on goal-directed communication with neural agents. Current MARL models typically assume the presence of a centralized rewarding signal that is accessible to all agents during training. This assumption can be realistic for certain scenarios such as agents learning to communicate to play soccer (all agents can perceive the score of the game). However, it is not for other conditions such as teaching where agents have asymmetrical affordances and knowledge, and where communication is a means for a more knowledgeable agent (teacher) to guide a less knowledgeable agent (student) towards the goal. Fig. 3.11 illustrates how the ABP differs from MARL communication and IRL setups.

The ABP is in fact a computational implementation of an experimental semiotics investigation: the Coconstruction Game (Vollmer et al., 2014). In their experiment, the builder and the architect are humans. They are located in separate rooms. The architecture has a picture of a target lego block structure while the builder is seated at a table in front of a set of lego blocks. The architect monitors the builder workspace via a camera (video stream) and must send messages to the builder until it manages to construct the structure. In order to prevent pre-existing communication systems from influencing the results of their studies, the architect uses a button box with neutral symbols (designed to minimize the presence of biases such as color or shape, so as to avoid the attribution of preexisting meanings). We explore the ABP in chapter 5. More specifically, we propose an algorithmic solution to it in a construction environment like the Coconstruction Game. We investigate the key learning dynamics in terms of mutual information between mes-

sages and actions and show that agents can agree on a communication protocol enabling them to generalize to new constructions never seen during training.

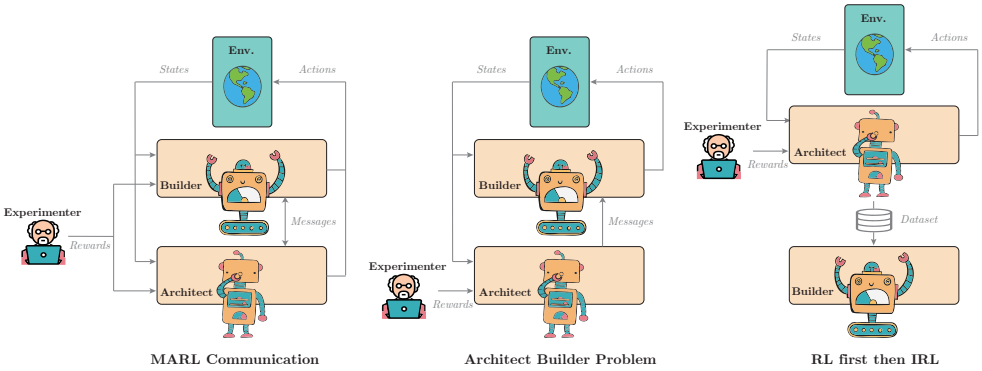


Figure 3.11: **The Architect Builder Problem and how it differs with respect to other AI paradigms.** Conversely to MARL communication (a), in ABP, the architect cannot act in the environment and the builder never perceives the reward (b). Because the architect cannot act in the environment, it is impossible to frame the problem as an RL and then IRL problem (c).

3.3 Self-organisation of Trajectories: the Open-ended Skill Acquisition Problem

The second part of this manuscript studies the role of cultural conventions in the self-organization of developmental trajectories of artificial agents. In this section, we propose to introduce our contributions in a similar fashion to what we did for the formation of cultural conventions. We first present a typology of the computational models enabling artificial agents to self-organize repertoires of skills before introducing our contributions.

3.3.1 Computational Models of the Formation of Skill Repertoires with Autotelic RL

Beyond modeling language formation, developmental AI aims to model how children learn skills in general. In this section, we propose a computational framework that addresses the challenge of self-organizing developmental trajectories and the open-ended learning of skill repertoires. The framework, referred to as autotelic RL or developmental RL, is a combination of developmental approaches and reinforcement learning (see the definition of autotelic in Sec. 1.1.1). It builds on *intrinsic motivations* (IMs) to enable agents to learn to represent, generate, select, and solve their own problems. To provide a comprehensive understanding of the framework, we first present a typology of intrinsic motivation approaches in developmental AI, followed by a presentation of the autotelic learning problem and its solution with autotelic agents.

Intrinsic Motivations in Developmental AI

Developmental AI aims to model children learning and, thus, takes inspiration from the mechanisms underlying autonomous behaviors in humans. Most of the time, humans are not motivated by external rewards but spontaneously explore their environment to discover and learn about what is around them. This behavior is driven by *intrinsic motivations* (IMs) a set of brain processes that motivate humans to explore for the mere purpose of experiencing novelty, surprise or learning progress (Berlyne, 1966; Gopnik et al., 1999; Kidd & Hayden, 2015a; Oudeyer & Smith, 2016; Gottlieb & Oudeyer, 2018).

The integration of IMs into artificial agents thus seems to be a key step towards autonomous learning agents (Schmidhuber, 1991; Kaplan & Oudeyer, 2007). In developmental robotics, this approach enabled sample efficient learning of high-dimensional motor skills in complex robotic systems (Santucci et al., 2020), including locomotion (Baranes & Oudeyer, 2013; Martius et al., 2013), soft object manipulation (Rolf & Steil, 2013; Nguyen & Oudeyer, 2014), visual skills (Lonini et al., 2013) and nested tool use in real-world robots (Forestier et al., 2022). Most of these seminal approaches leverage *population-based* optimization algorithms, i.e. non-parametric models trained on (outcome, policy) pairs. These methods train separate policies for each goal, often demonstrate limited generalization capabilities, and cannot easily handle high-dimensional perceptual spaces.

Recently, we have been observing a convergence between developmental robotics and deep RL, forming a new domain that we propose to call *developmental reinforcement*

learning as a subfield of developmental AI. Indeed, RL researchers now incorporate fundamental ideas from the developmental robotics literature in their own algorithms, and reversely developmental robotics learning architectures are beginning to benefit from the generalization capabilities of deep RL techniques. These convergences can mostly be categorized in two ways depending on the type of intrinsic motivation (IMs) being used (Oudeyer & Kaplan, 2007):

- **Knowledge-based IMs** are about prediction. They compare the situations experienced by the agent to its current knowledge and expectations and reward it for experiencing dissonance (or resonance). This family includes IMs rewarding prediction errors (Schmidhuber, 1991; Pathak et al., 2017), novelty (Bellemare et al., 2016; Burda et al., 2019; Raileanu & Rocktäschel, 2020), surprise (Achiam & Sastry, 2017), negative surprise (Berseth et al., 2019), learning progress (Lopes et al., 2012; Kim et al., 2020) or information gains (Houthooft et al., 2016), see a review in (Linke et al., 2020). This type of IM is often used as an auxiliary reward to organize the exploration of agents in environments characterized by sparse rewards. It can also be used to facilitate the construction of world models (Lopes et al., 2012; Kim et al., 2020; Sekar et al., 2020).
- **Competence-based IMs**, on the other hand, are about control. They reward agents to solve self-generated problems, to achieve self-generated goals. In this category, agents need to represent, select and master self-generated goals. As a result, competence-based IMs were often used to organize the acquisition of repertoires of skills in task-agnostic environments (Baranes & Oudeyer, 2010, 2013; Santucci et al., 2016; Forestier & Oudeyer, 2016; Nair et al., 2018b; Warde-Farley et al., 2019; Colas et al., 2019a; Blaes et al., 2019; Pong et al., 2020).

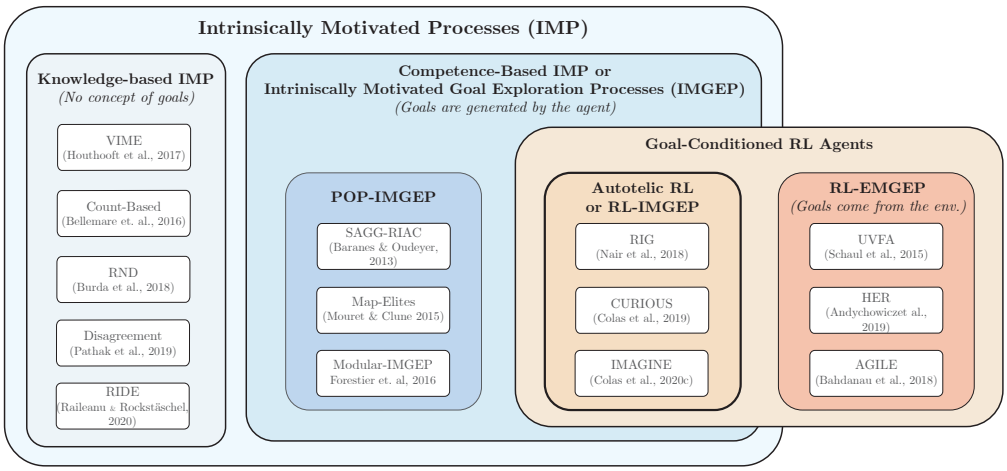


Figure 3.12: A typology of intrinsically-motivated and/or goal-conditioned RL approaches. POP-IMGEP, RL-IMGEP and RL-EMGEP refer to population-based intrinsically motivated goal exploration processes, RL-based IMGEP and RL-based externally motivated goal exploration processes respectively.

Fig. 3.12 proposes a visual representation of intrinsic motivations approaches (knowledge-based IMs vs competence-based IMs or IMGEPs) and RL approaches (intrinsically vs externally motivated). RL algorithms using *knowledge-based* IMs (on the left) leverage ideas from developmental robotics to solve standard RL problems. On the other hand, algo-

rithms using competence-based IMs organize exploration around self-generated goals and can be seen as targeting a developmental robotics problem: the *open-ended formation of skill repertoires*. *Intrinsically Motivated Goal Exploration Processes* (IMGEP) is the family of autotelic algorithms that bake competence-based IMs into learning agents (Forestier et al., 2022). IMGEP agents generate and pursue their own goals as a way to explore their environment, discover possible interactions, and build repertoires of skills. This framework emerged from the field of developmental robotics (Oudeyer & Kaplan, 2007; Baranes & Oudeyer, 2009a, 2010; Rolf et al., 2010) and originally leveraged population-based learning algorithms (POP-IMGEP) (Baranes & Oudeyer, 2009b, 2013; Forestier & Oudeyer, 2016; Forestier et al., 2022). The intersection between IMGEP and multi-goal RL are autotelic RL algorithms or RL-IMGEP. They train agents to generate and pursue their own goals by training goal-conditioned policies. They contrast with RL-EMGEP agents which do not generate their own goals and rely on externally provided ones.

The Autotelic Learning problem

In the *autotelic learning problem* or the *open-ended formation of skill repertoires*, the agent is set in an open-ended environment without any pre-defined goal and needs to acquire a repertoire of skills. Here, we use the definition of skill provided in Sec. 2.3, i.e. the association of a goal embedding z_g and the policy to reach it Π_g. A repertoire of skills is thus defined as the association of a repertoire of goals $\mathcal{G}$ with a goal-conditioned policy trained to reach them $\Pi_{\mathcal{G}}$. The intrinsically motivated skills acquisition problem can now be modeled by a reward-free MDP $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0\}$ that only characterizes the agent, its environment and their possible interactions. Just like children, agents must be autotelic, i.e. they should learn to represent, generate, pursue, and master their own goals. Fig. 3.13 illustrates the key difference between multi-goal RL (Sec. 2.3) and autotelic RL. In multi-goal RL an experimenter provides goals and rewards to the agent.

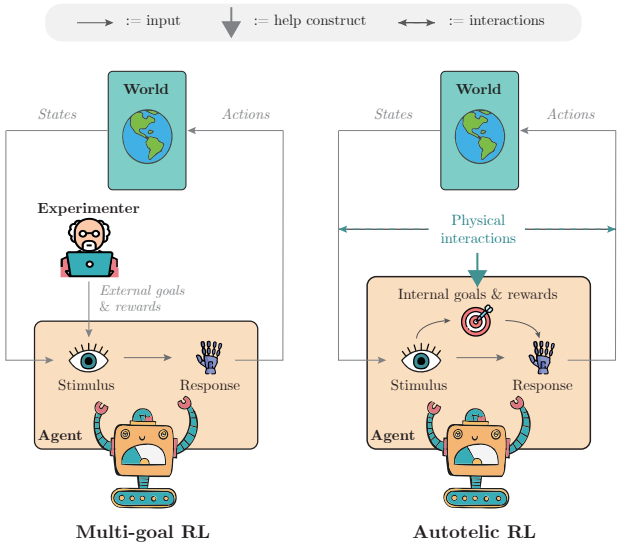


Figure 3.13: **Multi-goal RL vs Autotelic RL.** In autotelic RL, agents learn to represent, generate, pursue and master their own goals. Goals are thus internal to the agent while multi-goal RL relies on an external experimenter providing goals and rewards.

Evaluating Autotelic Agents

Evaluating agents is often trivial in reinforcement learning. Agents are trained to maximize one or several pre-coded reward functions — the set of possible interactions is known in advance. One can measure generalization abilities by computing the agent's success rate on a held-out set of testing goals. One can measure exploration abilities via several metrics such as the count of task-specific state visitations.

In contrast, autotelic agents evolve in open-ended environments and learn to represent and form their own set of skills. In this context, the space of possible behaviors might quickly become intractable for the experimenter, which is perhaps the most interesting feature of such agents. For these reasons, designing evaluation protocols is not trivial. The evaluation of such systems raises similar difficulties as the evaluation of task-agnostic content generation systems like Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) or self-supervised language models (Devlin et al., 2019; Brown et al., 2020b). In both cases, learning is *task-agnostic* and it is often hard to compare models in terms of their outputs (e.g. comparing the quality of GAN output images, or comparing output repertoires of skills in autotelic agents).

- **Measuring exploration:** one can compute task-agnostic exploration proxies such as the entropy of the visited state distribution, or measures of state coverage (e.g. coverage of the high-level x-y state space in mazes) (Florensa et al., 2018). Exploration can also be measured as the number of interactions from a set of *interesting* interactions defined subjectively by the experimenter (interactions with objects as we do in chapter 8).
- **Measuring generalization:** The experimenter can define a set of relevant target goals and prevent the agent from training on them. Evaluating agents on this held-out set at test time provides a measure of generalization (Ruis et al., 2020), although it is biased towards what the experimenter assesses as *relevant* goals.
- **Measuring transfer learning:** The intrinsically motivated exploration of the environment can be seen as a pre-training phase to bootstrap learning in a subsequent downstream task. In the downstream task, the agent is trained to achieve externally-defined goals. We report its performance and learning speed on these goals. This is akin to the evaluation of self-supervised language models, where the reported metrics evaluate performance in various downstream tasks (Brown et al., 2020b).
- **Opening the black-box:** Investigating internal representations learned during intrinsically motivated exploration is often informative. One can investigate properties of the goal generation system (e.g. does it generate out-of-distribution goals?), investigate properties of the goal embeddings (e.g. are they disentangled?). One can also look at the learning trajectories of the agents across learning, especially when they implement their own curriculum learning (Florensa et al., 2018; Colas et al., 2019a; Blaes et al., 2019; Pong et al., 2020; Akakzia et al., 2021a).
- **Measuring robustness:** Autonomous learning agents evolving in open-ended environment should be robust to a variety of properties than can be found in the real-world. This includes very large environments, where possible interactions might vary in terms of difficulty (trivial interactions, impossible interactions, interactions

whose result is stochastic thus prevent any learning progress). Environments can also include distractors (e.g. non-controllable objects) and various forms of non-stationarity. Evaluating learning algorithms in various environments presenting each of these properties allows to assess their ability to solve the corresponding challenges.

Autotelic RL Agents

Autotelic agents (or RL-IMGEP) are intrinsically motivated versions of goal-conditioned RL algorithms. They need to be equipped with mechanisms to represent and generate their own goals in order to solve the autotelic learning problem. Concretely, this means that, in addition to the goal-conditioned policy, they need to learn: 1) to represent goals g by compact embeddings z_g; 2) to represent the support of the goal distribution, also called *goal space* $\mathcal{Z}_\mathcal{G} = \{z_g\}_{g\in\mathcal{G}}$; 3) a goal distribution from which targeted goals are sampled $\mathcal{D}(z_g)$; 4) a goal-conditioned reward function $\mathcal{R}_\mathcal{G}$. This four modules are illustrated in Fig. 3.14. In practice, only a few architectures tackle the four learning problems above. Indeed, simple autotelic agents assume pre-defined goal representations (1), the support of the goals distribution (2) and goal-conditioned reward functions (4). As autotelic architectures tackle more of the 4 learning problems, they become more and more advanced. As we will see in the following sections, many existing works in goal-conditioned RL can be formalized as autotelic agents by including goal sampling mechanisms *within the definition of the agent.*

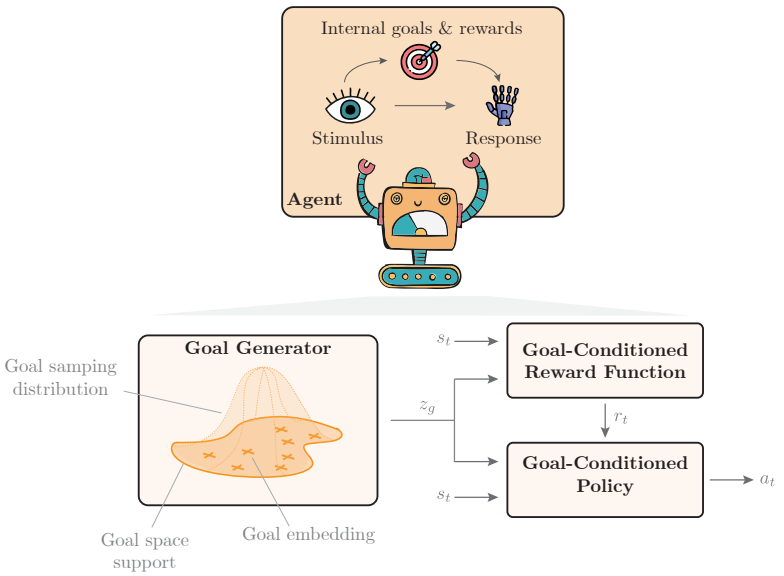


Figure 3.14: Representation of the different learning modules in an autotelic agent.

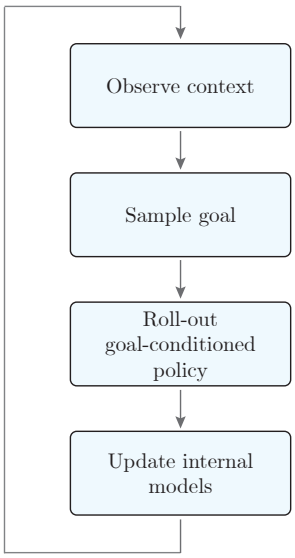
Algorithm 2 details the pseudo-code of RL-IMGEP algorithms. Starting from randomly initialized modules and memory, RL-IMGEP agents enter a standard RL interaction loop. They first observe the context (initial state), then sample a goal from their goal sampling policy. Then starts the proper interaction. Conditioned on their current goal embedding,

they act in the world so as to reach their goal, i.e. to maximize the cumulative rewards generated by the goal-conditioned reward function. After the interaction, the agent can update all its internal models. It learns to represent goals by updating its goal embedding function and goal-conditioned reward function, and improves its behavior towards them by updating its goal-conditioned policy.

Algorithm 2: Autotelic Agent with RL-IMGEP

Require: environment $\mathcal{E}$

Initialize empty memory $\mathcal{M}$,goal-conditioned policy $\Pi_{\mathcal{G}}$, goal-conditioned reward $R_{\mathcal{G}}$,goal space $\mathcal{Z}_{\mathcal{G}}$, goal sampling policy GS.
loop
  Get initial state: $s_0 \leftarrow \mathcal{E}$.reset()
  Sample goal embedding $z_g = GS(s_0, \mathcal{Z}_{\mathcal{G}})$.
  Execute a roll-out with $\Pi_g = \Pi_{\mathcal{G}}(\cdot \mid z_g)$
  Store collected transitions $\tau = (s, a, s')$ in $\mathcal{M}$.
  Sample a batch of B transitions: $\mathcal{M} \sim \{(s, a, s')\}_B$.
  Perform Hindsight Relabelling $\{(s, a, s', z_g)\}_B$.
  Compute internal rewards $r = R_{\mathcal{G}}(s, a, s' \mid z_g)$.
  Update policy $\Pi_{\mathcal{G}}$ via RL on $\{(s, a, s', z_g, r)\}_B$.
  Update goal representations $\mathcal{Z}_{\mathcal{G}}$.
  Update goal-conditioned reward function $R_{\mathcal{G}}$.
  Update goal sampling policy GS.
end loop
return $\Pi_{\mathcal{G}}, R_{\mathcal{G}}, \mathcal{Z}_{\mathcal{G}}$



General RL-IMGEP loop

Most RL-EMGEP approaches use pre-defined goal representations where goal spaces and associated rewards are pre-defined by the engineer and are part of the task definition (see our topology of goal representation in Sec. 2.3). On the other hand, autotelic agents actually need to learn these goal representations. While individual goals are represented by their embeddings and associated reward functions, representing multiple goals also requires the representation of the *support* of the goal space, i.e. how to represent the collection of *valid goals* that the agent can sample from, see Fig. 3.14. In addition to constructing a goal space, autotelic agents must sample goals within that space to actually explore the world. The next two sections address the questions of how to learn goal representations and how to select goals.

How to Learn Goal Representations?

Learning Goal Embeddings. Some approaches assume the pre-existence of a goal-conditioned reward function, but learn to represent goals by learning goal embeddings. This is the case of language-based approaches, which receive rewards from the environment (thus are RL-EMGEP), but learn goal embeddings jointly with the policy during policy learning (Hermann et al., 2017a; Chan et al., 2019a; Jiang et al., 2019a; Bahdanau et al., 2019c; Hill et al., 2020a; Cideron et al., 2020c; Lynch & Sermanet, 2020). When goals are target images, goal embeddings can be learned via generative models of states,

assuming the reward to be a fixed distance metric computed in the embedding space (Nair et al., 2018b; Florensa et al., 2019; Pong et al., 2020; Nair et al., 2020).

Learning reward functions. A few approaches go even further and learn their own goal-conditioned reward function. In the domain of image-based goals, Venkattaramanujam et al. (2019); Hartikainen et al. (2020) learn a distance metric estimating the square root of the number of steps required to move from any state s_1 to any s_2 and generates internal signals to reward agents for getting closer to their target goals. Warde-Farley et al. (2019) learn a similarity metric in the space of controllable aspects of the environment that is based on a mutual information objective between the state and the goal state s_g. This method is reminiscent of *empowerment* methods Mohamed & Rezende (2015); Gregor et al. (2016); Achiam et al. (2018); Eysenbach et al. (2019); Dai et al. (2020); Sharma et al. (2020); Choi et al. (2021). Empowerment methods aim at maximizing the mutual information between the agent's actions or goals and its experienced states. Recent methods train agents to develop a set of skills leading to maximally different areas of the state space. Agents are rewarded for experiencing states that are easy to discriminate, while a discriminator is trained to better infer the skill z_g from the visited states. This discriminator acts as a skill-specific reward function.

In the domain of language goals, Bahdanau et al. (2019a); Colas et al. (2020b) learn language-conditioned reward functions from an expert dataset or from language descriptions of autonomous exploratory trajectories respectively. However, the AGILE approach from Bahdanau et al. (2019a) does not generate its own goals.

Learning the supports of goal distributions. Finally, to represent collections of goals, agents need to represent the support of the goal distribution — which embeddings correspond to valid goals and which do not. To this end, most approaches consider a pre-defined, bounded goal space in which any point is a valid goal (e.g. target positions within the boundaries of a maze, target block positions within the gripper's reach) (Schaul et al., 2015; Andrychowicz et al., 2017a; Nair et al., 2018a; Plappert et al., 2018; Colas et al., 2019a; Blaes et al., 2019; Lanier et al., 2019; Ding et al., 2019; Li et al., 2020). However, not all approaches assume pre-defined goal spaces. However, some approaches use the set of previously experienced representations to form the support of the goal distribution (Veeriah et al., 2018; Akakzia et al., 2021a; Ecoffet et al., 2021). In Florensa et al. (2018), a Generative Adversarial Network (GAN) is trained on past representations of states ($\varphi(s)$) to model a distribution of goals and thus its support. In the same vein, approaches handling image-based goals usually train a generative model of image states based on Variational Auto-Encoders (VAE) to model goal distributions and support (Nair et al., 2018b; Pong et al., 2020; Nair et al., 2020). In both cases, valid goals are the one generated by the generative model.

How to Select Goals?

Once autotelic agents have constructed a goal support inside a goal space, they need to specify a goal selection policy. Although agents can sample their goal space uniformly, informed goal selection can be a way for agents to organize their learning curriculum automatically.

Automatic curriculum learning (ACL). Applied for goal selection, ACL is a mechanism that organizes goal sampling so as to maximize long-term performance improvement (distal objective). As this objective is usually not directly differentiable, curriculum learning techniques usually rely on a proximal objective. Proxies include *intermediate difficulty* (Sukhbaatar et al., 2018; Campero et al., 2021; Zhang et al., 2020), *novelty-diversity* (Warde-Farley et al., 2019; Pong et al., 2020; Pitis et al., 2020; Kovač et al., 2020; Fang et al., 2021) or *medium-term learning progress* (Baranes & Oudeyer, 2013; Moulin-Frier et al., 2014; Forestier & Oudeyer, 2016; Fournier et al., 2018, 2021; Colas et al., 2019a; Blaes et al., 2019; Portelas et al., 2020a). Interested readers can refer to Portelas et al. (2020b), which present a broader review of ACL methods.

Hierarchical reinforcement learning (HRL). HRL can be used to guide the sequencing of goals (Dayan & Hinton, 1993a; Sutton et al., 1998, 1999; Precup, 2000). In HRL, a high-level policy is trained via RL or planning to generate sequence of goals for a lower level policy so as to maximize a higher-level reward. This allows to decompose tasks with long-term dependencies into simpler sub-tasks. Low-level policies are implemented by traditional goal-conditioned RL algorithms (Levy et al., 2018; Röder et al., 2020) and can be trained independently from the high-level policy (Kulkarni et al., 2016; Frans et al., 2018) or jointly (Levy et al., 2018; Nachum et al., 2018; Röder et al., 2020).

Summary

In this section, we presented the autotelic RL framework. This paradigm, at the intersection of developmental robotics and standard AI technics, builds intrinsically motivated agents that generate and pursue their own problems. Autotelic agents fall in the category of competence-based IMs. Unlike standard multi-goal RL agents (presented in Sec. 2.3) that rely on externally provided goals, autotelic agents discover and learn to represent their own goals from their experience of the physical world. This ability to develop in symbiosis with the physical world is reminiscent of Piaget's developmental psychology (Piaget, 1952) which highlights children's ability to shape their learning trajectories with respect to their sensory-motor experience of the world. We propose a classification of autotelic RL-IMGEP approaches in Tab. 3.1.

Approach	Goal Type	Goal Rep.	Reward Function	Goal sampling strategy
RL-IMGEPs that assume goal embeddings and reward functions				
Fournier et al. (2018)	Target features (+tolerance)	Pre-def	Pre-def	LP-Based
HAC Levy et al. (2018)	Target features	Pre-def	Pre-def	HRL
HIRO Nachum et al. (2018)	Target features	Pre-def	Pre-def	HRL
CURIOUS Colas et al. (2019a)	Target features	Pre-def	Pre-def	LP-based
CLIC Fournier et al. (2021)	Target features	Pre-def	Pre-def	LP-based
CWYC Blaes et al. (2019)	Target features	Pre-def	Pre-def	LP-based + surprise
GO-EXPLORE Ecoffet et al. (2021)	Target features	Pre-def	Pre-def	Novelty
NGU Badia et al. (2020b)	Objectives balance	Pre-def	Pre-def	Uniform
AGENT 57 Badia et al. (2020a)	Objectives balance	Pre-def	Pre-def	Meta-learned
DECSTR Akakzia et al. (2021a)	Binary problem	Pre-def	Pre-def	LP-based
SLIDE Fang et al. (2021)	Skill index	Pre-def	Pre-def	Novelty (PCG)
XLAND OEL Stooke et al. (2021)	Binary problem	Pre-def	Pre-def	Intermediate difficulty
RL-IMGEPs that learn their goal embedding and assume reward functions				
RIG Nair et al. (2018b)	Target features (images)	Learned (VAE)	Pre-def	From VAE prior
GOALGAN Florensa et al. (2018)	Target features	Pre-def + GAN	Pre-def	Intermediate difficulty
Florensa et al. (2019)	Target features (images)	Learned (VAE)	Pre-def	From VAE prior
SKEW-FIT Pong et al. (2020)	Target features (images)	Learned (VAE)	Pre-def	Diversity
SETTER-SOLVER Racanière et al. (2019)	Target features (images)	Learned (Gen. model)	Pre-def	Uniform difficulty
MEGA Pitis et al. (2020)	Target features (images)	Learned (VAE)	Pre-def	Novelty
CC-RIG Nair et al. (2020)	Target features (images)	Learned (VAE)	Pre-def	From VAE prior
AMIGO Campero et al. (2021)	Target features (images)	Learned (with policy)	Pre-def	Adversarial
GRIMGEP Kovač et al. (2020)	Target features (images)	Learned (with policy)	Pre-def	Diversity and ALP
Full RL-IMGEPs				
DISCERN Warde-Farley et al. (2019)	Target features (images)	Learned (with policy)	Learned (similarity)	Diversity
DIAYN Eysenbach et al. (2019)	Discrete skills	Learned (with policy)	Learned (discriminability)	Uniform
Hartikainen et al. (2020)	Target features (images)	Learned (with policy)	Learned (distance)	Intermediate difficulty
Venkattaramanujam et al. (2019)	Target features (images)	Learned (with policy)	Learned (distance)	Intermediate difficulty
IMAGINE Colas et al. (2020b)	Binary problem (language)	Learned (with reward)	Learned	Uniform + Diversity
VGCRL Choi et al. (2021)	Target features	Learned	Learned	Empowerment

Table 3.1: **A classification of autotelic RL-IMGEP approaches.** The classification groups algorithms depending on their degree of autonomy: 1) RL-IMGEPs that rely on pre-defined goal representations (embeddings and reward functions); 2) RL-IMGEPs that rely on pre-defined reward functions but learn goal embeddings and 3) RL-IMGEPs that learn complete goal representations (embeddings and reward functions). For each algorithm, we report the type of goals being pursued, whether goal embeddings are learned, whether reward functions are learned, and how goals are sampled. We mark in bold algorithms that use a developmental approach and explicitly pursue the intrinsically motivated skills acquisition problem.

3.3.2 Problem Definition

In the second part of this manuscript, we will investigate the role of cultural conventions in the self-organization of agents' developmental trajectories. To do so, we will extend the autotelic RL framework presented in the previous section (Sec. 3.3.1). Complementing the Piagetian approach of autotelic RL and inspired by the literature deriving from Vygotsky (1934)'s theory of child development, we propose a new framework called *Vygotskian Autotelic AI*.

The initial contribution of Part II is conceptual. It proposes to draw the contours of an AI framework where agents leverage pre-existing cultural conventions to transform their learning abilities. It is important to note that, in contrast with the first portion of this research, this investigation will examine the scenario of artificial agents using pre-established cultural conventions to organize their developmental trajectories, disregarding any negotiation of protocols or multi-agent dynamics. In Vygotksian Autotelic agents do not only interact with the physical world surrounding them but with social partners. They are immersed in a (rich) sociocultural environment. An illustration of the difference between autotelic RL and Vygotskian autotelic RL is provided in Fig. 3.15.

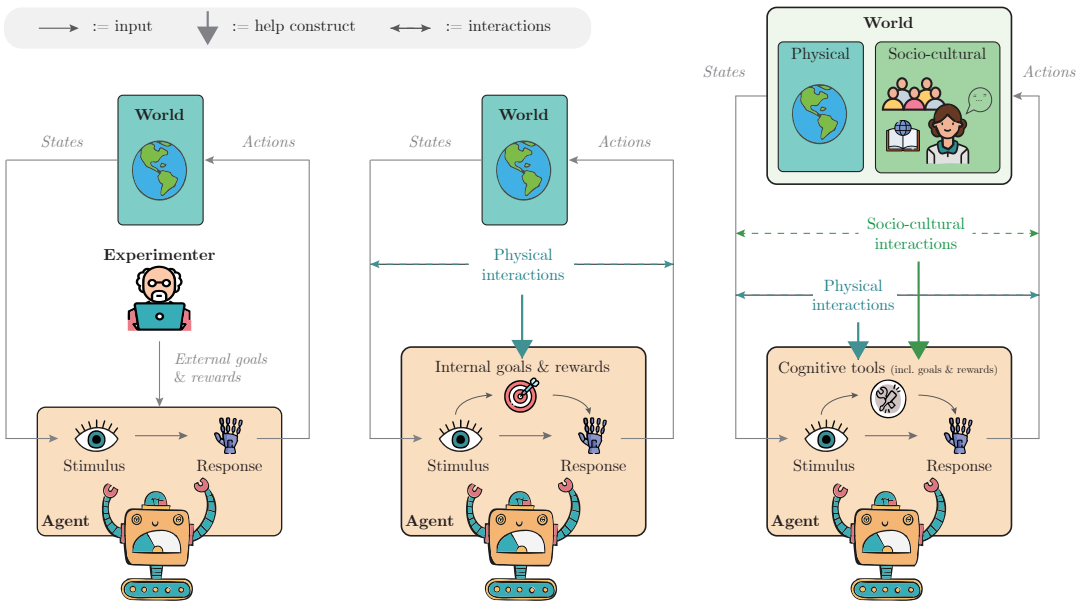


Figure 3.15: From multi-goal RL to autotelic RL to Vygotskian autotelic RL. RL defines an agent experiencing the state of the world as stimuli and acting on that world via actions. Multi-goal RL (a): goals and associated rewards come from pre-engineered functions and are perceived as sensory stimuli by the agent. Autotelic RL (b): agents build internal goal representations from interactions between their intrinsic motivations and their physical experience (Piagetian view). Vygotskian autotelic RL (c): agents internalise physical and socio-cultural interactions into *cognitive tools*. Here, *cognitive tools* refer to any self-generated representation that mediates stimulus and actions: self-generated goals, explanations, descriptions, attentional biases, visual aids, mnemonic tricks, etc.

To benefit from sociocultural and linguistic conventions, Vygotksian autotelic agents

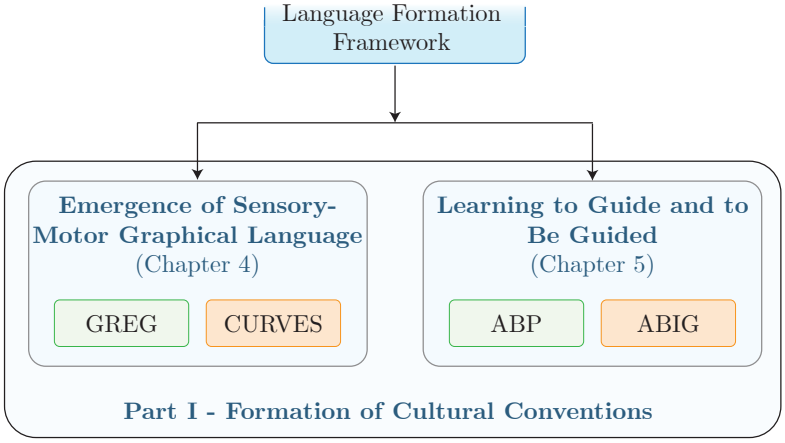
need to ground them into their own sensory-motor modalities. They need to extract the structure of language and align it with their sensory-motor experience. As will be discussed, agents do not merely extract the structure of language, but instead assimilate the entire convention, generating an internal representation of the social partner with whom they are interacting. This internal model can be called at any time to generate plans in an autotelic fashion for instance. We argue that this Vygotskian framework can palliate autotelic agents' serious limitations in terms of goal diversity, exploration, generalization, or skill composition. To back this claim, we present two experimental contributions displaying how agents can ground complex spatiotemporal language (chapter 7) and how they can use language as a cognitive tool to generate goals in curiosity-driven exploration (chapter 8). Both of these experimental contributions will leverage the *Playground* environment: a socio-physical environment made of a variety of objects with different properties and a simulated social partner providing linguistic descriptions of interesting interactions.

In chapter 7 we will equip Vygotkian artificial agents with transformer neural network architectures to enable them to align their experience of the world with linguistic descriptions provided by a surrogate social partner. More specifically, we will investigate the impact of relational inductive biases on a specific extractive module of vyogtskian agents: the reward function; and show how those inductive biases can lead to better systematic generalization.

Finally, in chapter 8 we will detail the implementation of IMAGINE: a Vygotskian autotelic agent that converts linguistic descriptions given by a social partner into targetable goals. We will show that IMAGINE can learn productive modules and internalize socio-cultural conventions in order to leverage language productivity and systematic generalization to grow an open-ended repertoire of skills in a creative way.

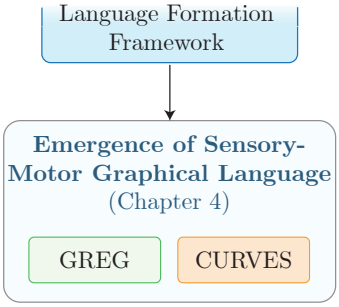
Part I

Formation of Cultural Conventions



Chapter 4

Self-Organization of a Sensory-motor Graphical Language



Contents

In our study of the formation of cultural conventions between artificial agents, we first investigate whether artificial agents can develop a shared language in an ecological setting where communication relies on a *sensory-motor channel*. To this end, we extend the setup of neural language games described in Sec. 3.2.1 and introduce the Graphical Referential Game (GREG). In the GREG, a speaker must produce a graphical utterance to name a visual referent object consisting of combinations of MNIST digits while a listener has to select the corresponding object among distractor referents, given the produced message. The utterances are drawing images produced using dynamical motor primitives combined with a sketching library. To tackle GREG we present CURVES: **C**ontrastive **U**tterance-**R**eferent associati**VE** **S**coring, a multimodal contrastive deep learning mechanism that represents the energy (alignment) between named referents and utterances generated

through gradient ascent on the learned energy landscape. We demonstrate that CURVES not only succeed at solving the GREG but also enable agents to self-organize a language that generalizes to feature compositions never seen during training. In addition to evaluating the communication performance of our approach, we also explore the structure of the emerging language. Specifically, we show that the resulting language forms a coherent lexicon that is shared between agents and that basic compositional rules on the graphical productions could not explain the compositional generalization

4.1 Motivations

As we described in Sec. 3.2.1, most approaches to language games have considered only idealized symbolic communication channels based on discrete tokens (Lazaridou et al., 2017; Mordatch & Abbeel, 2018; Chaabouni et al., 2021) or fixed-size sequences of word tokens (Havrylov & Titov, 2017; Portelance et al., 2021). This predefined means of communication is motivated by language's discrete and compositional nature. But how can this specific structure emerge during vocalization or drawing, for instance? Although fundamental in the investigation of the origin of language (Dessalles, 2000; Cheney & Seyfarth, 2005; Oller et al., 2019), this question seems to be neglected by recent approaches to Language Games (Moulin-Frier & Oudeyer, 2020). We, therefore, propose to study how communication could emerge between agents producing and perceiving continuous signals with a constrained *sensory-motor system*.

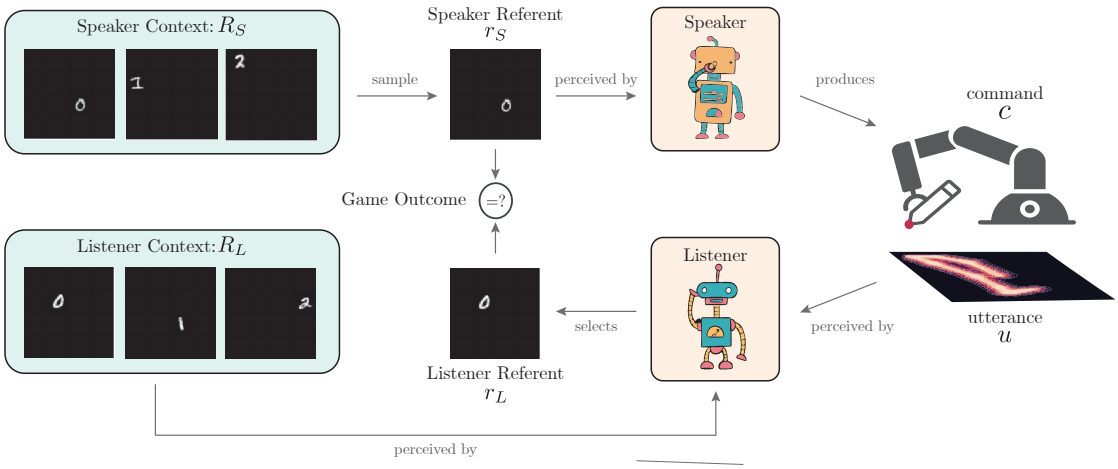


Figure 4.1: **The Graphical Referential Game:** During an instantiation of the game, the speaker's goal is to produce a motor command c that will yield an utterance u in order to denote a referent r_S sampled from a context $\tilde{R}_S$. Following this step, the listener needs to interpret the utterance in order to guess the referent it denotes among a context $\tilde{R}_L$. The game is a success if the listener and the speaker agree on the referent ($r_L \equiv r_S$).

Such continuous constrained systems have been used in the cognitive science literature as models of sign production to study the self-organization of speech in artificial systems (de Boer, 2000; Oudeyer, 2006; Moulin-Frier et al., 2015). In this chapter, we focus on a drawing sensory-motor system producing graphical signs. The sensory-motor system is made of Dynamical Motor Primitives (DMPs) (Schaal, 2006) combined with

a sketching system (Mihai & Hare, 2021a) enabling the conversion of motor commands into images. Drawing systems have the advantage of producing 2D trajectories interpretable by humans while preserving the non-linear properties of speech models, which were shown to ease the discretization of the produced signals (Stevens, 1989; Moulin-Frier et al., 2015). We introduce the *Graphical Referential Game*: a variation of the original referential game, where a *Speaker* agent (top of Fig. 4.1) has to produce a graphical *utterance* given a single target *referent* while a *Listener* agent (bottom of Fig. 4.1) has to select an element among a context made of several referents, given the produced utterance (agents alternate their roles). In this setting, we first investigate whether a population of agents can converge on an efficient communication protocol to solve the graphical language game. Then, we evaluate the coherence and compositional properties of the emergent language, since it is one of the main characteristics of human languages.

Early language game implementations (Steels, 1995b, 2001) achieve communication convergence by using contrastive methods to update association tables between object referents and utterances (see Fig. 3.7 of chapter 3). While recent works use deep learning methods to target high-dimensional signals they do not explore contrastive approaches. Instead, they model interactions as a multi-agent reinforcement learning problem where utterances are actions, and agents are optimized with policy gradients, using the outcomes of the games as the reward signal (Lazaridou et al., 2017). In the meantime, recent models leveraging contrastive multimodal mechanisms such as CLIP (Radford et al., 2021) have achieved impressive results in modeling associations between images and texts. Combined with efficient generative methods (Ramesh et al., 2021), they can compose textual elements that are reflected in image form as the composition of their associated visual concepts. Inspired by these techniques, we propose CURVES: Contrastive Utterance-Referent associatiVE Scoring, an algorithmic solution to the graphical referential game. CURVES relies on two mechanisms: 1) The contrastive learning of an energy landscape representing the alignment between utterances and referents and 2) the generation of utterances that maximize the energy for a given target referent. We evaluate CURVES in two instantiations of the graphical referential game: one with symbolic referents encoded by one-hot vectors and another with visual referents derived from the multiple MNIST digits (LeCun et al., 1998). We show that CURVES converges to a shared graphical language that enables a population of agents not only to name complex visual referents but also to name new referent compositions that were never encountered during training.

Scope

The idea of using a sensory-motor system to study the emergence of forms of combinatoriality in language dates back to methods investigating the origins of digital vocalization systems (de Boer, 2000; Oudeyer, 2005; Zuidema & De Boer, 2009). Such studies were conducted in the context of imitation games at the level of phonemes to observe the formation of speech utterances (syllables, words) that were systematically composed from lower-level meaningless elements (phonemes). This corresponded to the first level of compositionality within the notion of duality of patterning (Hockett & Hockett, 1960). Yet, these works did not consider referential games and did not study agents' ability to

compose meaningful words to denote referents, i.e. they did not address the second level of the duality of patterning.

One of the goals of emergent communication research is to develop machines that can interact with humans. As a result, a variety of referential game approaches ensure that the emergent language is as close to natural language. This can be achieved by adding a supervised image captioning objective to encourage agents to use natural language in order to solve their communicative tasks (Havrylov & Titov, 2017; Lazaridou et al., 2017). Other methods use constraints such as memory restrictions (Kottur et al., 2017) to act as an information bottleneck to increase interpretability and compositionality. While we purposefully chose a graphical sensory-motor system to ease the visualization of the emerging language, we do not inject prior knowledge or pressures to facilitate the emergence of an iconic language. Our produced utterances are completely arbitrary. This fundamentally differentiates our work from Mihai & Hare (2021b) that trains agents to communicate via sketches replicating the visual referents they name. Note also that their drawing setup does not include dynamical motor primitives and utterances are directly optimized in image space. They, moreover, allow gradients to back-propagate from listener to speaker while we use a decentralized approach. Finally, they do not consider contrastive learning. To our knowledge, CURVES is the first contrastive deep-learning algorithm successfully applied to a referential game.

There is a large body of work exploring the factors that promote compositionally in emerging languages (Kottur et al., 2017; Li & Bowling, 2019; Rodríguez Luna et al., 2020; Ren et al., 2020; Chaabouni et al., 2020; Gupta et al., 2020). In this context, a crucial question is how to actually measure it in the first place (Mu & Goodman, 2021). To this end, (Choi et al., 2018b) proposes to measure communicative performances on unseen compositions of known objects as a way to evaluate compositionality. However, it has been shown that a good performance in this test may be achieved without leveraging any actual compositionality in language (Andreas, 2019; Chaabouni et al., 2020). Thus, others instead compute topographic similarities (Brighton & Kirby, 2006), measuring the correlation between distances in the utterance space (distance between signs) and distances in the referents space (such as the cosine similarity between the embeddings of objects) (Lazaridou et al., 2018). In this contribution we propose to do both and study 1) the generalization to unseen combinations of abstract features and 2) topographic measures based on the Hausdorff distances between utterances denoting composition and utterances denoting isolated features.

Specific Contributions

The specific contributions introduced in this chapter are:

- The Graphical Referential Game (GREG): a variation of the referential language game to study the formation of signs from a graphical sensory-motor system.
- CURVES: an algorithmic solution to GREG, consisting of a contrastive multimodal encoder coupled with a generative model enabling the emergence of a graphical language.
- A study of CURVES's generalization performances on compositions of features never seen during training in a simplified control setting and a more perceptually chal-

lenging one.
- A complementary analysis of the structure of the emerging graphical language measuring lexicon coherence and compositionality scores derived from the Haussdorf distance.

4.2 The Graphical Referential Games

We consider a group of two agents playing a fixed number of referential games, each time alternating their roles (speaker or listener). During a game, we first present a context R of n objects, called referents to a speaker S and a listener L. At the beginning of each game, the target $r^\star \in R$ is assigned to the speaker. Given this target referent $r^\star$, S produces an utterance (u) to designate it. Based on the produced utterance u, L selects a referent $(\hat{r})$ in R. The game outcome o is a success if the selected referent $(\hat{r})$ matches the target $r^\star$.

The setup

Referents. Referents are compositions of orthogonal vector features (one-hot vectors). Given a set of m orthogonal features F_m, we define the set of all possible referents as $\mathcal{R}_m = \{\sum_{f\in S} f | S \subseteq F_m\}$. The subset of referents made of exactly k features are thus: $\mathcal{R}_m^k = \{\sum_{f\in S} f | S \subseteq F_m, |S| = k\}$. In our experiments, we fix $m = 5$.

From these orthogonal referents, we propose to generate objects made of digit images sampled from the MNIST dataset (LeCun et al., 1998). More precisely, we define the stochastic mapping $\Phi : \mathcal{R}_m \to \tilde{\mathcal{R}}_m$ that maps each feature $f \in F_m$ to a digit class in the MNIST dataset. For each feature in a referent, we sample a random instance from the corresponding class and randomly place it on a 4×4 grid such that no number overlap. Note that the listener and speaker can perceive different realizations of Φ, in this case, we say that they see different *perspectives* of the referents. More precisely, the speaker perceives the context R as $\tilde{R}_S$ and its target $r^\star$ as $r^\star_S$. Similarly, the listener perceives the context R as $\tilde{R}_L$ and selects a referent $\hat{r}$ among it.

We use this formalism to instantiate three settings of the Graphical Referential Game (GREG):

- *one-hot*: where referents are one-hot vectors $r \in \mathcal{R}_m$.
- *visual-shared*: where referents are MNIST digits $r \in \tilde{\mathcal{R}}_m$ and agents share the same perspective: $\tilde{R}_S = \tilde{R}_L$.
- *visual-unshared* where referents are MNIST digits $r \in \tilde{\mathcal{R}}_m$ and agents have different perspectives of referents in their contexts $\tilde{R}_S \neq \tilde{R}_L$.

Sensory-motor drawing system. Utterances are produced by a sensory-motor system $M : \mathbb{R}^m \to \mathcal{U} \subset \mathbb{R}^{D\times D}$ mimicking an arm drawing sketches displayed in Fig. 4.2(a). The arm motion is derived from Dynamical Motor Primitives (DMPs) (Schaal, 2006). The DMP is parametrized by a command vector $c \in \mathbb{R}^{20}$. Each of the x and y positions of the pen is controlled by a DMP starting at the center of the image and parameterized by 10 weights. These weights are the parameters of the motion of a one-dimensional oscillator that generates a smooth drawing trajectory T made of 10 coordinates $T = \{v_i\}_{i=0,\dots,9}$.

The parameters of the two DMPs are given in Suppl. table A.1. The trajectory is then fed to a Differentiable Sketching model (Mihai & Hare, 2021a) generating an $D \times D$ image (in our implementation, $D = 52$).

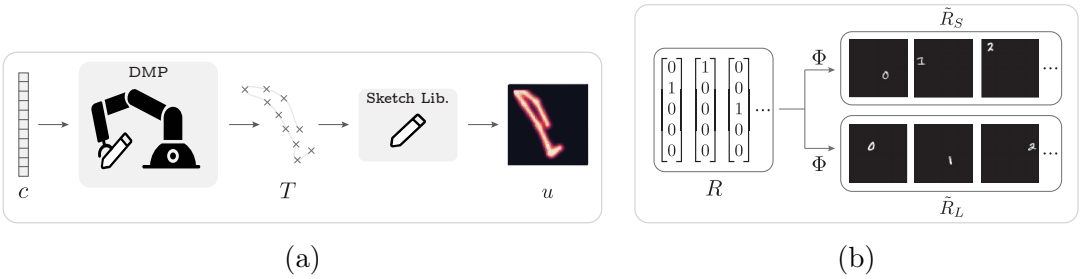


Figure 4.2: (a) **Sketching sensory-motor system**: The sensory-motor system imitates a robotic arm drawing a sketch on a 2D plan. DMPs first convert a continuous command c into a sequence of coordinates T. This trajectory is then rendered as a 52×52 graphical utterance thanks to a differentiable sketching library. (b) **Referent transformation:** An example of a one-hot context R being transformed into two contexts $\tilde{R}_S$ and $\tilde{R}_L$ by the stochastic transformation Φ. The two contexts are different perspectives of the same objects.

Objectives

In this study, we aim to answer the three following questions:

1. What are agents' communicative performances in the GREG? Are agents able to solve the game? Are they able to generalize to compositional referents?
2. Are the emergent signs coherent? Do agents produce the same utterances to denote the same referents?
3. Are the emergent signs compositional? Are there compositional rules in the production of signs naming compositional referents? [1]

Are agents able to solve the GREG? To answer the first question, we will monitor the communicative performance of agents on both training and testing referents. The training referents consist of a single feature: $\mathcal{R}_{\text{train}} = \mathcal{R}_5^1$ while the testing referents consists of two features: $\mathcal{R}_{\text{test}} = \mathcal{R}_5^2$. For visual examples of compositional referents, see Suppl. Section A.1.2.

Are the emergent signs coherent? To measure coherence we propose to use a similarity measure based on the Hausdorff distance. Haussdorf distance is known to capture geometric features of trajectories, in particular, their shape (Besse et al., 2015). The Hausdorff distance d_H is the maximum distance from any coordinate in a trajectory to the closest coordinate in the other: $d_H(T_1, T_2) = \max\{\sup_{v \in T_1} d(v, T_2), \sup_{v' \in T_2} d(T_1, v')\}$. In particular, we compute the following metrics.

- Agent Coherence (A-coherence): For a given referent r with the same perspective for all agents, measure the mean pairwise similarity between each agent's utterance.

[1] Note that the ability to perform compositional generalization (question 1) and the presence of compositional structure in utterances (question 3) are two separate investigations.

- Perspective Coherence (P-coherence): For a given agent and a given referent r, measure the mean pairwise similarity between utterances produced from different perspectives
- Referent Coherence (R-coherence): For a given agent, measure the mean pairwise similarity between utterances produced for different referents.

Are the emergent signs compositional? To measure the compositionally of the utterances, we introduce a topographic score based on the Hausdorff distance ρ. ρ quantifies how an utterance denoting a compositional referent made of feature i and j ($u(r_{ij})$) is actually closer to the utterances denoting isolated features $u(r_i)$ or $u(r_j)$ than the utterance naming other compositional referents ($u(r_{xy})$, $x \neq i, y \neq j$). For a detailed derivation of metric ρ, see Suppl. Section A.1.3.

4.3 CURVES: Contrastive Utterance-Referent associatiVE Scoring

CURVES is an energy-based approach that relies on two mechanisms:

1. The contrastive learning of an energy landscape $E(r, u)$, defined as the cosine similarity between utterance and referent embeddings.
2. The generation of an utterance that maximizes the energy for a given target referent $r^\star_S$.

Agents modules and interactions.

Each agent $A \in \{A_1, A_2\}$ perceives utterances and referents using two distinct CNN encoders f_A (for referents) and g_A (for utterances)[2]. f_A and g_A map referents and utterances in a shared d-dimensional latent space: $f_A(\cdot, \theta_{fA}) : \mathcal{R}_m \to \mathbb{R}^d$ and $g_A(\cdot, \theta_{gA}) : \mathcal{U} \to \mathbb{R}^d$ such that $z_{rA} = f_A(r)$ and $z_{uA} = g_A(u)$, as displayed in Fig. 4.3(a). The agent then computes the energy landscape as: $E_A(r, u) = \cos(f_A(r), g_A(u))$.

A given referential game unfolds as follows. Agents have randomly attributed roles, for instance, A_1 is the speaker $A_1 \leftarrow S$ and A_2 is the listener $A_2 \leftarrow L$. The speaker is given a context $\tilde{R}_S$ and a target referent perceived as $r^\star_S$ to produce an utterance $\hat{u}$ intending to approach the utterance $u^\star$ that maximizes $E_S(r^\star_S, u)$. The listener observes $\hat{u}$ and selects referent $\hat{r}$ in context $\tilde{R}_L$ that maximizes $E_L = (r, \hat{u})$:

$$\begin{cases} \hat{u} \approx u^\star = \underset{u \in \mathcal{U}}{\operatorname{argmax}}\ E_S(r^\star_S, u) \\ \hat{r} = \underset{r \in \tilde{R}_L}{\operatorname{argmax}}\ E_L(r, \hat{u}) \end{cases} \tag{4.1}$$

The outcome of the game is then $o = \mathbb{1}_{[\hat{r}=r^\star]} - b$ where b is a baseline parameter representing the mean success across previous games.

[2] when referents are one-hot vectors f_A is a fully-connected network. Parameters for both encoders are given in Suppl. table A.2.

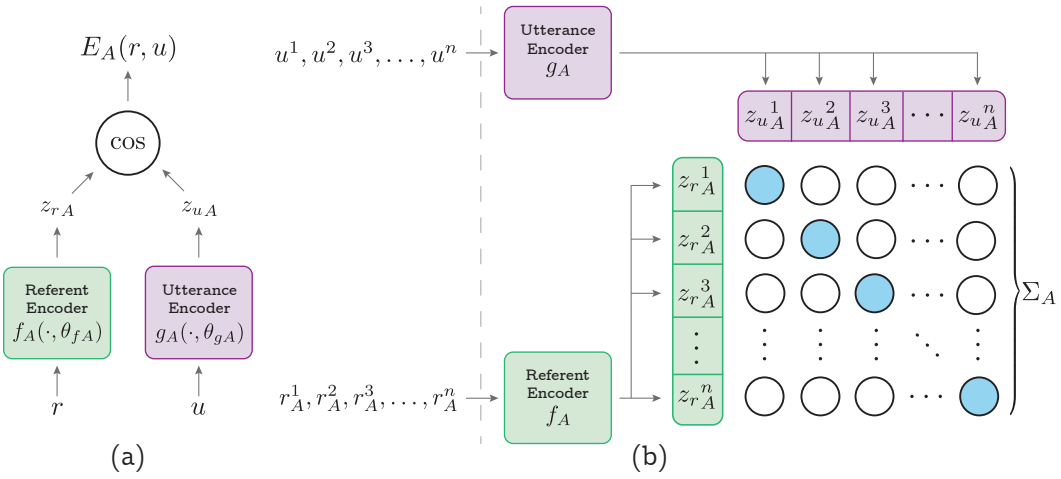


Figure 4.3: (a) **Agents's dual encoder architecture.** Referents and utterances are mapped to a share latent space. The energy between a referent r and an utterance u is computed as the cosine similarity between their respective embeddings. (b) **Cosine similarity matrix update from collected samples.** Agents compute the energy for all referents and utterances they collected to form the squared matrix Σ_A. During contrastive updates agents maximize blue circles and minimize white ones.

Contrastive representation learning in referential games.

For a given context R, agents are randomly assigned their roles and play $n = |R|$ games. During these n games, roles are fixed and the speaker agent successively selects each referent of the context $\tilde{R}_S$ as the target $r_S^\star$. During interactions, the speaker collects data $\{(r_S^i, u^i, o^i)\}_{i=1,\ldots,n}$ while the listeners observes $\{(u^i, r_L^i)\}_{i=1,\ldots,n}$. From the collected data each agent can compute the squared cosine similarity matrices Σ_A whose elements are $(\Sigma_A)_{i,j} = E_A(r_A^i, u^j)$ as shown in Fig. 4.3(b). Contrastive updates are then performed using the objective J_A that applies *Cross Entropy* (CE) on the i-th row and i-th column of Σ_A.

$$J_A(\Sigma_A, i) = \frac{CE((\Sigma_A)_{i,1:n}, e_i) + CE((\Sigma_A)_{1:n,i}, e_i)}{2} \tag{4.2}$$

e_i being a one-hot vector of size n with value 1 at index i. Depending on the role of the agent, J_A is instantiated either as J_S (speaker) or J_L (listener). Thus, the speaker updates its representation using the outcomes o_i of the games (reinforcing the successful associations while decreasing the unsuccessful ones):

$$\underset{\theta_{f_S}, \theta_{g_S}}{\text{minimize}} \sum_{i=1}^{n} o_i J_S(\Sigma_S, i) \tag{4.3}$$

On the other hand, the listener needs to make sure that the selection matches the speaker's referent (Steels, 2015) and hence always increases associations (no matter the games' outcomes):

$$\underset{\theta_{f_L}, \theta_{g_L}}{\text{minimize}} \sum_{i=1}^{n} J_L(\Sigma_L, i) \tag{4.4}$$

Note that in Eq. 4.4, r_L^i is the target referent perceived by the listener. This means that, at the end of the game, the speaker indicates the referent (as perceived by the listener)

that they named. As reviewed in Sec. 3.3.1, this retroactive pointing mechanism was employed in both early language game implementations (Steels, 1995a) and more recent ones (Lazaridou et al., 2017; Chaabouni et al., 2020; Portelance et al., 2021).

Speaker's utterance optimization.

We distinguish two utterance generation strategies:

- The descriptive generation: in which the speaker agent only considers the target referent $r_S^\star$ to produce an utterance that maximizes the cosine similarity between the embeddings of $r_S^\star$ and an utterance produced by our sensory system $u = M(c)$ from motor command c. Since M is fully differentiable, we inject the sensory-motor constraint in equation 4.1 and seek for the optimal motor command $c^\star$ using gradient ascent:

$$c^\star = \underset{c \in \mathbb{R}^p}{\operatorname{argmax}}\ E(r_S^\star, M(c)) \tag{4.5}$$

- The discriminative generation: in which the speaker also perceives the context $\tilde{R}_S$ during production. This is achieved by finding the motor command that minimizes the cross entropy given a target referent $r_S^\star$ and its context $\tilde{R}_S$:

$$c^\star = \underset{c \in \mathbb{R}^p}{\operatorname{argmin}}\ CE(\sigma_S, e_{r_S^\star}) \tag{4.6}$$

  where σ_S is the vector with coordinates $\sigma_{Si} = [E(r^i, M(c))]_{r^i \in \tilde{R}_S}$ and $e_{r_S^\star}$ is the one-hot vector of size $|\tilde{R}_S|$ with value 1 at the position of $r_S^\star$ in $\tilde{R}_S$. This discriminative generation process is only used at test time when investigating CURVES's generalization capabilities.

4.4 Experiments

4.4.1 Communicative Performance

In all three settings of the Graphical Referential Game (one-hot, visual-shared, and visual-unshared), agents succeed and achieve a perfect training success rate of 1.

Generalization to compositional referents.

Table 4.1 exposes the generalization performances of agents evaluated on referents $r \in \mathcal{R}_5^2$. During an evaluation, the context is exhaustive and contains all the combinations of 2 features: $|R| = 10$. We compare the success rates to a *random* baseline where the listener always selects the referent $\hat{r}_L$ randomly no matter the utterance ($\text{SR}_{\text{random}} = 0.1$). We also introduce a *1-feature* baseline where the speaker produces an utterance u that only denotes one of the two features contained in $r_S^\star$ and the listener randomly selects one of the four combinations containing the communicated feature ($\text{SR}_{\text{1-feat}} = 0.25$).

Referents	**Descriptive** SR	**Discriminative** SR
One-hot	0.99 ± 0.01	0.99 ± 0.01
Visual-shared	0.57 ± 0.04	0.56 ± 0.03
Visual-unshared	0.39 ± 0.02	0.40 ± 0.02

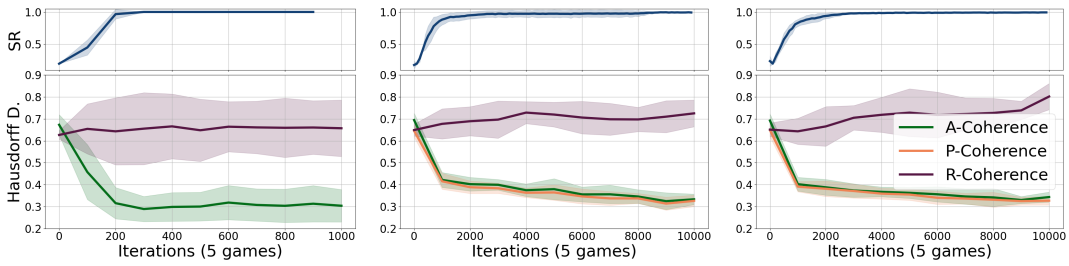
Table 4.1: **Generalization performances.** Success rates evaluated on exhaustive context $|R| = 10$ with referents $r \in \mathcal{R}_5^2$ for both generative (Eq. 4.5) and discriminative (Eq.4.6) utterance generation.

The success rates for all referent types are significantly higher than the baseline values suggesting that agents are indeed able to communicate about compositional referents. Generalization performances are nearly perfect with one-hot referents but they decrease in visual settings. This performance gap can be explained by the extra difficulty of adding inter-perspective variability to the multi-agent interaction dynamic during the contrastive learning of referent representations. The better success rates obtained in auto-learning (where a single agent plays both the speaker and the listener roles) provided in Suppl. Section A.2.1 seem to corroborate this hypothesis. Surprisingly, we observe that success rates for descriptive (Eq. 4.5) and discriminative (Eq.4.6) generation are very similar. This suggests that optimizing utterances so as to minimize their energy between non-targeted compositional referents ($r \in R, r \neq r^\star$) does not improve generalization performances.

4.4.2 Structure of the Emergent Language

Coherence

Fig. 4.4 displays the evolution of the inter-agent (A), inter-perspective (P), and inter-referent (R) coherence during training. A group starts to converge and succeed at the game when inter-agent and inter-perspective coherence distances decrease. This correlation is proof of emergent communication as it indicates that agents start agreeing on signs to denote referents. The constant (for one-hot referent) and increasing (for visual referents) values of the R-coherence suggest that agents use distinct signs to name referents.



(a) (b) (c)

Figure 4.4: **Training success rate (SR) and Coherence distances** (a) one-hot referents (b) visual-shared referents (c) visual-unshared referents.

As displayed in Fig. 4.5, the language used by agents self-organizes around five distinct symbols. It is important to note that this self-organization arises from the production of continuous signals with no explicit communication of the five categories of visual referents. Other visualizations for one-hot and shared visual referents are available in Suppl. Section A.2.2. We also provide illustrations of P-coherence in Suppl. Section A.2.3.

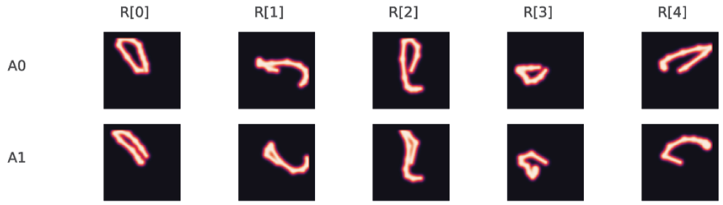
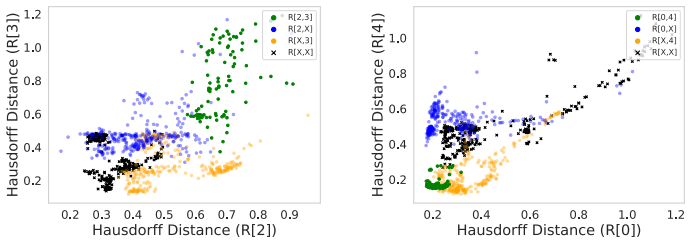


Figure 4.5: **Instance of an emerging lexicon.** Utterances are produced by a pair of agents trained with unshared perspectives (1 seed). The perspective for each referent is chosen randomly.

Compositionality

In Sec. 4.4.1, we showed that agents achieve a near-perfect success rate at naming compositions of one-hot features at test time. Is this successful communication reflected by a compositional structure in the produced signs? To investigate this question we propose the topographic maps associated with their topographic scores in Fig. 4.6.



(a) $\rho = -0.401$ (b) $\rho = 0.147$

Figure 4.6: **Topographic map examples for a single seed in one-hot referents setting**. Each utterance names a compositional referent and is colored in blue if it contains feature i ($R[i, X]$), orange if it contains feature j ($R[X, j]$), green if it contains both ($R[i, j]$), and black if it contains none ($R[X, X]$). (a) Corresponding to the worst topographic score $\rho = -0.401$ (combination of feature $i = 2$ and $j = 3$) (b) Corresponding to the best topographic score $\rho = 0.147$ (combination of feature $i = 0$ and $j = 4$).

Each point in a topographic map is an utterance naming a compositional referent $r \in \mathcal{R}_5^2$ and has coordinate $(d_H(u(r_i), \cdot), d_H(u(r_j), \cdot))$. Utterances at the bottom left of the topographic maps are therefore simultaneously close to the two utterances naming the isolated features. All the topographic maps are available in Fig. A.10 of Suppl. Section A.2.4. They show that for a minority of compositions (3 out of 10), the utterances naming the composition of two features are not close in Haussdorf distance to the

utterances naming the two isolated features ($\rho < 0$). This indicates that proximity in Haussdorf distance is not a necessary condition for agents to generalize on compositional referents. The matrix of composition provided in Fig. 4.7 illustrate that it is indeed very difficult to infer a composition rule from the generated utterances.

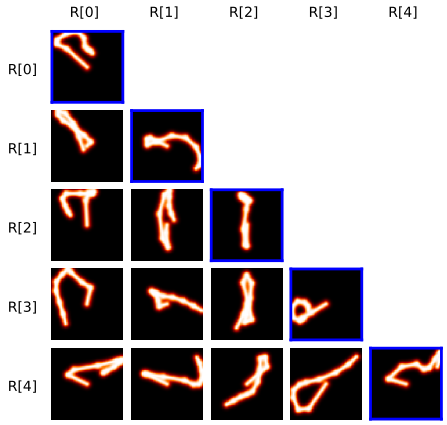


Figure 4.7: **Matrix of compositions.** Blue frames represent utterances generated for a perspective in $\mathcal{R}_5^1$, other utterance denote the corresponding compositions in $\mathcal{R}_5^2$

Despite the fact that we cannot perceive the compositional structure of emerging signs, the internal representations of agents seem to leverage compositional mechanisms. The t-snes provided in Fig. 4.8 shows that the embeddings for both compositional referents and the utterances naming them are close to their constituents.

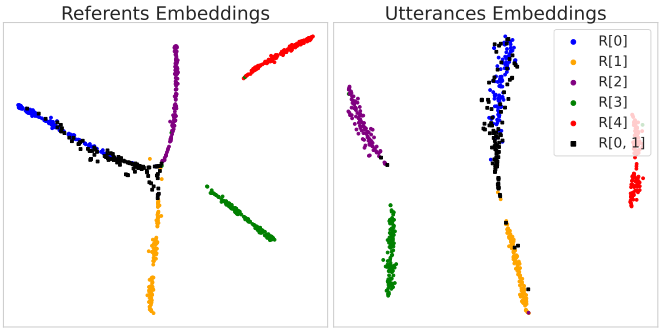


Figure 4.8: **T-sne of utterance and referent embeddings.** Embeddings are computed for 100 perspectives in the visual-unshared setting. Additional t-snes are provided in Suppl. Section A.2.6.

Conclusion

If the Haussdorf distance does not enable us to identify compositional rules in the production of utterances, it is particularly relevant for describing their coherence. This paper, therefore, provides the first step toward understanding the mechanisms at hand

for the emergence of structure in self-organizing languages. The structural analysis we present sheds light on the importance of studying ecological systems. By performing ablation of the DMP in the sensory-motor system and considering a speaker agent directly optimizing a randomly initialized image, one can observe that agents directly optimizing utterances in pixel space can negotiate a successful communication protocol (as indicated in table 4.2). However, the absence of structure in the resulting lexicon (illustrated in figure 4.9) prevents us from using our coherence and topographic scores to analyze it.

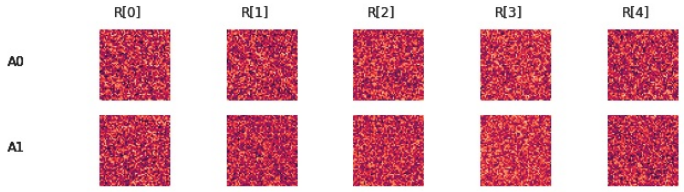


Figure 4.9: **Emerging lexicon without motion primitives.** Utterances naming referents with unshared perspectives.

	$\mathbf{SR}_{\text{train}}$	$\mathbf{SR}_{\text{test}}$
One-hot	0.99 ± 0.01	0.96 ± 0.02
Visual-shared	0.99 ± 0.01	0.55 ± 0.03
Visual-unshared	0.99 ± 0.01	0.41 ± 0.02

Table 4.2: **Training and generalization success without DMPs.** Utterances are generated in descriptive mode, and visual referents are seen from different perspectives.

4.5 Discussion and Future Work

In this chapter, we formalized GREG: a new ecological referential game where two agents must communicate via a continuous sensory-motor system imitating a robotic arm drawing sketches. To tackle GREG, we propose CURVES: a contrastive representation learning algorithm inspired by early language game contrastive implementation that scales to high dimensional signals. CURVES allows a group of two agents two converge on a shared graphical language in contexts where referents are one-hot vectors or images of MNIST digits. The representations that agents learn enable them to communicate about compositional referents never encountered during training. If the Haussdorf distance illustrates that emergent signs are coherent, it does not capture compositionality among them. Potential real-world experiments could be carried out to further investigate the ability of the Hausdorff metric to capture the structural compositionality of emergent signs. Such experiments could include the implementation of a graphical referential game among human participants, in which they will be requested to intentionally produce compositional signs. The data collected from this experiment would enable us to verify whether the Hausdorff topography metric can capture the structural compositionality of signs, by

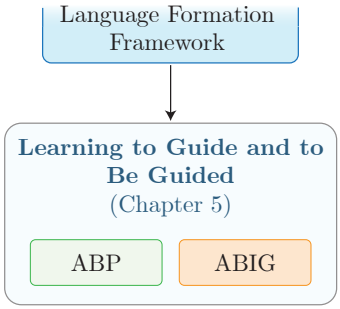
verifying whether compositional signs generated by humans are indeed located in the bottom left corner of our topographic maps.

Absence of baseline comparisons. The continuous nature of the utterances prevents us from comparing CURVES to the standard REINFORCE approach introduced by Lazaridou et al. (2018) and used in the majority of deep learning models of emergent communication. Indeed, REINFORCE considers a discrete space for utterances. To our knowledge, the only approach considering continuous and graphical utterances is provided by Mihai & Hare (2021b). However, using their implementation as a baseline is not applicable in our setting since their procedure allows gradients to propagate between agents which is not a realistic assumption to analyze the emergence of decentralized communication.

Perspectives. Future work may leverage our ecological setup and algorithmic solution to experiment with and test a variety of hypotheses that influence structures in self-organizing sing systems. An analysis of the impact of the sensory-motor constraints on the topology of graphical signs could for instance provide valuable insight into the ecological factors facilitating the emergence of a compositional graphical language. Inspired by work on the cultural evolution of language (Kirby, 2001), our setup can also serve as a basis to investigate and visualize the impact of other factors such as population dynamic or cognitive abilities of agents (with varying memory or perceptual systems). Finally, CURVES is agnostic to the modality used to represent utterances. As such, it could tackle other sensory-motor systems. The central element of CURVES lies in the contrastive learning of utterance-referent associations. In our implementation, we optimize utterances by maximizing this energy via gradient ascent. Much like CLIP opened many avenues for multi-modal generation, we could plug in more complex generative strategies such as diffusion models (Rombach et al., 2021; Saharia et al., 2022).

Chapter 5

Learning to Guide and to Be Guided in the Architect-Builder Problem



Contents

In contrast to the preceding chapter, which examines the self-organization of cultural conventions in the context of sensory-motor constraints in the classical language (or referential) game, the present chapter proposes to investigate the emergence of goal-directed communication between artificial agents in a novel setting. More specifically,

we study the collaboration between a *builder* – which performs actions but ignores the goal of the task, i.e. has no access to rewards – and an *architect* which guides the builder towards the goal of the task. This setting fundamentally differs from the standard MARL communication setup (presented at the end of Sec. 3.2.1) in which the reward function is provided to all agents.

In this new setting, the agents need to simultaneously learn a task while at the same time evolving a shared communication protocol. Ideally, such learning should only rely on high-level communication priors and be able to handle a large variety of tasks and meanings while deriving communication protocols that can be reused across tasks. Experimental Semiotics research has demonstrated human proficiency in learning from a priori unknown instructions and meanings. This study draws inspiration from Experimental Semiotics and introduces the Architect-Builder Problem (ABP). In this asymmetrical setting, an architect must learn to guide a builder toward constructing a specific structure. The architect knows the target structure but cannot act in the environment and can only send arbitrary messages to the builder. The builder on the other hand can act in the environment, but receives no rewards nor has any knowledge about the task, and must learn to solve it relying only on the messages sent by the architect. Crucially, the meaning of messages is initially not defined nor shared between the agents but must be negotiated throughout learning. Under these constraints, we propose Architect-Builder Iterated Guiding (ABIG), a solution to the Architect-Builder Problem where the architect leverages a learned model of the builder to guide it while the builder uses self-imitation learning to reinforce its guided behavior. To palliate to the non-stationarity induced by the two agents concurrently learning, ABIG structures the sequence of interactions between the agents into interaction frames. We analyze the key learning mechanisms of ABIG and test it in a 2-dimensional instantiation of the ABP where tasks involve grasping cubes, placing them at a given location, or building various shapes. In this environment, ABIG results in a low-level, high-frequency, guiding communication protocol that not only enables an architect-builder pair to solve the task at hand, but that can also generalize to unseen tasks.

5.1 Motivations

Humans have a remarkable ability to teach and learn from each other, which allows knowledge and skills to be shared and refined across generations. Even in situations where there is no shared language or common ground, such as a parent teaching a baby how to stack blocks during play, people can teach and be taught. Experimental Semiotics (Galantucci & Garrod, 2011), a line of work that studies the forms of communication that people develop when they cannot use pre-established ones, reveals that humans can even teach and learn without direct reinforcement signals, demonstrations, or shared communication protocols. Vollmer et al. (2014) for example investigate a co-construction (CoCo) game experiment where an architect must rely only on arbitrary instructions to guide a builder toward constructing a structure made of Lego blocks. In this experiment, both the task of building the structure and the meanings of the instructions – through which the architect guides the builder – are simultaneously learned throughout interactions. Are artificial agents capable of developing such cultural conventions?

As a first step toward this research direction, we draw inspiration from the CoCo game and propose the *Architect-Builder Problem* (ABP): an interactive learning setting that models agents' interactions with *Markov Decision Processes* (Puterman, 2014) (MDPs). In the ABP learning has to occur in a social context through observations and communication, in the absence of direct imitation or reinforcement (Bandura & Walters, 1977). Specifically, the constraints of the ABP are:

1. the builder has absolutely no knowledge about the task at hand (no reward and no prior on the set of possible tasks);
2. the architect can only interact with the builder through communication signals (cannot interact with the environment or provide demonstrations), and
3. the communication signals have no pre-defined meanings (nor belong to a set of known possible meanings).

(1) sets this work apart RL (Sec. 2.1) and even MARL (Sec. 2.4) where explicit rewards are available to all agents. (2) implies the absence of teleoperation or third-person demonstrations and thus distinguishes the ABP from IL (Sec. 2.2). Finally, (3) prevents the architect from relying on a fixed communication protocol since the meanings of instructions must be negotiated. Artificial agents exploiting pre-defined cultural conventions will be explored in part II of this manuscript.

These three constraints make ABP an appealing setting to investigate *Human-Robot Interaction* (HRI) (Goodrich & Schultz, 2008) problems where "a learner tries to figure out what a teacher wants them to do" (Grizou et al., 2013; Cederborg & Oudeyer, 2014). Specifically, the challenge of *Brain Computer Interfaces* (BCI), where users use brain signals to control virtual and robotic agents in sequential tasks (Katyal et al., 2014; de-Bettencourt et al., 2015; Mishra & Gazzaley, 2015; Muñoz-Moldes & Cleeremans, 2020; Chiang et al., 2021), is well captured by the ABP. In BCIs, (3) is identified as the calibration problem and is usually tackled with supervised learning to learn a mapping between signals and meanings. As this calibration phase is often laborious and impractical for users, current approaches investigate calibration-free solutions where the mapping is learned interactively (Grizou et al., 2014; Xie et al., 2021). Yet, these works consider that the user (i.e. the architect) is fixed, in the sense that it does not adapt to the agent (i.e. the builder) and uses a set of pre-defined instructions (or feedback) meanings that the agent must learn to map to signals. In our ABP formulation, however, the architect is dynamic and, as interactions unfold, must learn to best guide a learning builder by tuning the meanings of instructions according to the builder's reactions. In that sense, ABP provides a more complete computational model of agent-agent or human-agent interactions.

With all these constraints in mind, we propose Architect Builder Iterated Guiding (ABIG), an algorithmic solution to ABP where both agents are artificial agents. ABIG is inspired by the field of experimental semiotics and relies on two high-level interaction priors: *shared intent* and *interaction frames*. Shared intent refers to the fact that, although the builder ignores the objective of the task to fulfill, it will assume that its objective is aligned with the architect's. This assumption is characteristic of cooperative tasks and shown to be a necessary condition for the emergence of communication both in practice (Foerster et al., 2016; Cao et al., 2018) and in theory (Crawford & Sobel, 1982). Specifically, the builder should assume that the architect is guiding it toward

a shared objective. Knowing this, the builder must reinforce the behavior it displays when guided by the architect. We show that the builder can efficiently implement this by using imitation learning on its own guided behavior. Because the builder imitates itself, we call it self-imitation. The notion of *interaction frames* (also called *pragmatic frames*) states that agents that interact in sequence can more easily interpret the interaction history (Bruner, 1985; Vollmer et al., 2016). In ABIG, we consider two distinct interaction frames. These are stationary, meaning that when one agent learns, the other agent's behavior is fixed. During the first frame (the modeling frame), the builder is fixed and the architect learns a model of the builder's message-conditioned behavior. During the second frame (the guiding frame), the architect is fixed and the builder learns to be guided via self-imitation learning.

Specific Contributions

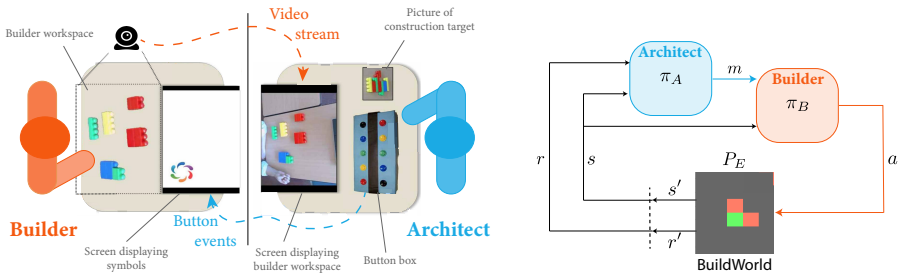
We show that ABIG results in a low-level, high-frequency, guiding communication protocol that not only enables an architect-builder pair to solve the task at hand, but can also be used to solve unseen tasks. **Our contributions are:**

- The Architect-Builder Problem (ABP), an interactive learning setting to study how artificial agents can simultaneously learn to solve a task and derive a communication protocol.
- Architect-Builder Iterated Guiding (ABIG), an algorithmic solution to the ABP.
- An analysis of ABIG's key learning mechanisms.
- An evaluation of ABIG on a construction environment where we show that ABIG agents evolve communication protocols that generalize to unseen harder tasks.
- A detailed analysis of ABIG's learning dynamics and impact on the mutual information between messages and actions (in the Supplementary Material).

5.2 The Architect-Builder Problem

The Architect-Builder Problem. We consider a multi-agent setup composed of two agents: an architect and a builder. Both agents observe the environment state s but only the architect knows the goal at hand. The architect cannot take actions in the environment but receives the environmental reward r whereas the builder does not receive any reward and has thus no knowledge about the task at hand. In this asymmetrical setup, the architect can only interact with the builder through a communication signal m sampled from its policy $\pi_A(m|s)$. These messages, that have no a priori meanings, are received by the builder which acts according to its policy $\pi_B(a|s,m)$. This makes the environment transition to a new state s' sampled from $P_E(s'|s,a)$ and the architect receives reward r'. Messages are sent at every time-step. The CoCo game that inspired ABP is sketched in Fig. 5.1(a) while the overall architect-builder-environment interaction diagram is given in Fig. 5.1(b). The differences between the ABP setting and the MARL and IRL settings are illustrated in Fig. B.2.

BuildWorld. We conduct our experiments in *BuildWorld*. BuildWorld is a 2D construction grid-world of size $(w \times h)$. At the beginning of an episode, the agent and N_b



(a) (b)

Figure 5.1: (a) **Schematic view of the CoCo Game (the inspiration for ABP).** The architect and the builder should collaborate in order to build the construction target while located in different rooms. The architecture has a picture of the target while the builder has access to the blocks. The architect monitors the builder workspace via a camera (video stream) and can communicate with the builder only through the use of 10 symbols (button events). (b) **Interaction diagram between the agents and the environment in our proposed ABP.** The architect communicates messages (m) to the builder. Only the builder can act (a) in the environment. The builder conditions its action on the message sent by the builder ($\pi_B(a|s,m)$). The builder never perceives any reward from the environment. A schematic view of the equivalent ABP problem is provided in Fig. B.1(b).

blocks are spawned at different random locations. The agent can navigate in this world and grasp blocks by activating its gripper while on a block. The action space $\mathcal{A}$ is discrete and include a "do nothing" action ($|\mathcal{A}| = 6$). At each time step, the agent observes its position in the grid, its gripper state as well as the position of all the blocks and if they are grasped ($|\mathcal{S}| = 3 + 3N_b$).

Tasks. BuildWorld contains 4 different training tasks:

1. 'Grasp': The agent must grasp any of the blocks;
2. 'Place': The agent must place any block at a specified location in the grid;
3. 'H-Line': The agent must place all the blocks in a horizontal line configuration;
4. 'V-Line':The agent must place all the blocks in a vertical line configuration.

BuildWorld also has a harder fifth testing task, '6-blocks-shapes', that consists of more complex configurations and that is used to challenge an algorithm's transfer abilities. For all tasks, rewards are sparse and only given when the task is completed.

This environment encapsulates the interactive learning challenge of ABP while removing the need for complex perception or locomotion. In the RL setting, where the same agent acts and receives rewards, this environment would not be very impressive. However, it remains to be shown that the tasks can be solved in the setting of ABP (with a reward-less builder and an action-less architect).

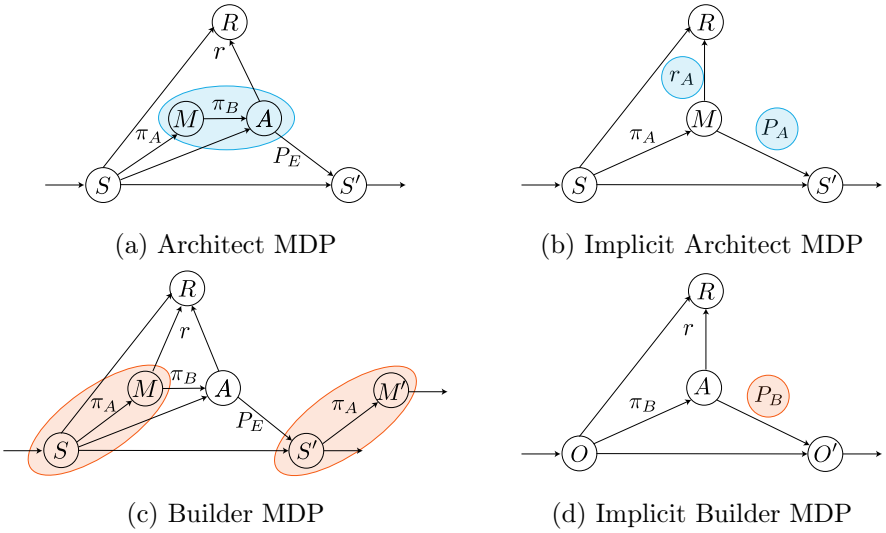
Communication. The architect guides the builder by sending messages m which are one-hot vectors of size $|\mathcal{V}|$ ranging from 2 to 72, see 5.4.5 for the impact of this parameter.

Additional Assumptions. In order to focus on the architect-builder interactions and the learning of a shared communication protocol, the architect has access to $P_E(s'|s,a)$ and to the reward function $r(s,a)$ of the goal at hand. This assumes that, if the architect were to act in the environment instead of the builder, it would be able to quickly figure

out how to solve the task. This assumption is compatible with the CoCo game experiment (Vollmer et al., 2014) where humans participants, and in particular the architects, are known to have such world models.

5.3 ABIG: Architect-Builder Iterated Guiding

5.3.1 Analytical Description



(a) Architect MDP (b) Implicit Architect MDP

(c) Builder MDP (d) Implicit Builder MDP

Figure 5.2: **Agent's Markov Decision Processes.** Highlighted regions refer to MDP coupling. (a) The architect's transitions and rewards are conditioned by the builder's policy π_B. (b) Architect's MDP where transition and reward models implicitly account for builder's behavior. (c-d) The builder's transition model depends on the architect's message policy π_A. The builder's learning signal r is unknown.

Agents-MDPs. In the Architect-Builder Problem, agents are operating in different, yet coupled, MDPs. Those MDPs depend on their respective point of view (see Figure 5.2). From the point of view of the architect, messages are actions that influence the next state as well as the reward (see Fig. 5.2 (a)). The architect knows the environment transition function $P_E(s'|s,a)$ and $r(s,a)$, the true reward function associated with the task that does not depend explicitly on messages. It can thus derive the effect of its messages on the builder's actions that drive the reward and the next states (see Fig. 5.2 (b)). On the other hand, the builder's state is composed of the environment state and the message, which makes estimating state transitions challenging as one must also capture the message dynamics (see Fig. 5.2 (c)). Yet, the builder can leverage its knowledge of the architect picking messages based on the current environment state. The equivalent transition and reward models, when available, are given below (see derivations in Suppl.

Section B.2).

$$\left.\begin{aligned} P_{A}(s'|s,m) &= \sum_{a\in\mathcal{A}} \tilde{\pi}_{B}(a|s,m) P_{E}(s'|a,s) \\ r_{A}(s,m) &= \sum_{a\in\mathcal{A}} \tilde{\pi}_{B}(a|s,m) r(s,a) \end{aligned}\right\} \quad \text{with} \quad \tilde{\pi}_{B}(a|s,m) \triangleq P(a|s,m) \tag{5.1}$$

$$P_{B}(s',m'|s,m,a) = \tilde{\pi}_{A}(m'|s') P_{E}(s'|s,a) \quad \text{with} \quad \tilde{\pi}_{A}(m'|s') \triangleq P(m'|s') \tag{5.2}$$

where subscripts A and B refer to the architect and the builder, respectively. $\tilde{x}$ denotes that x is unknown and must be approximated. From the builder's point of view, the reward – denoted $\tilde{r}$ – is unknown. This prevents the use of classical RL algorithms.

Shared Intent and Interaction Frames. It follows from Eq. (5.1) that, provided that it can approximate the builder's behavior, the architect can compute the reward and transition models of its MDP. It can then use these to derive an optimal message policy π_{A}^{*} that would maximize its objective:

$$\pi_{A}^{*} = \underset{\pi_{A}}{\operatorname{argmax}}\, G_{A} = \underset{\pi_{A}}{\operatorname{argmax}}\, \mathbb{E}[\sum_{t} \gamma^{t} r_{A,t}] \tag{5.3}$$

$\gamma \in [0,1]$ is a discount factor and the expectation can be thought of in terms of π_{A}, P_{A} and the initial state distribution. However, the expectation can also be though in terms of the corresponding trajectories $\tau \triangleq \{(s,m,a,r)_{t}\}$ generated by the architect-builder interactions. In other words, when using π_{A}^{*} to guide the builder, the architect-builder pair generates trajectories that maximizes G_{A}. The builder has no reward signal to maximize, yet, it relies on a shared intent prior and assumes that its objective is the same as the architect's one:

$$G_{B} = G_{A} = \mathbb{E}_{\tau}[\sum_{t} \gamma^{t} r_{A,t}] = \mathbb{E}_{\tau}[\sum_{t} \gamma^{t} \tilde{r}_{t}] \tag{5.4}$$

where the expectations are taken with respect to trajectories τ of architect-builder interactions. Therefore, under the shared intent prior, architect-builder interactions where the architect uses π_{A}^{*} to maximize G_{A} also maximize G_{B}. This means that the builder can interpret these interaction trajectories as demonstrations that maximize its unknown reward function $\tilde{r}$. Consequently, the builder can reinforce the desired behavior – towards which the architect guides it – by performing self-Imitation Learning[1] on the interaction trajectories τ.

Note that in Eq. (5.1), the architect's models can be interpreted as expectations with respect to the builder's behavior. Similarly, the builder's objective depends on the architect's guiding behavior. This makes one agent's MDP highly non-stationary and the agent must adapts its behavior if the other agent's policy changes. To palliate to this, agents rely on interaction frames which means that, when one agent learns, the other agent's policy is fixed to restore stationarity. The equivalent MDPs for the architect and the builder are respectively $\mathcal{M}_{A} = \langle \mathcal{S}, \mathcal{V}, P_{A}, r_{A}, \gamma \rangle$ and $\mathcal{M}_{B} = \langle \mathcal{S} \times \mathcal{V}, \mathcal{A}, P_{B}, \emptyset, \gamma \rangle$. Finally, $\pi_{A} : \mathcal{S} \mapsto \mathcal{V}$, $P_{A} : \mathcal{S} \times \mathcal{V} \mapsto [0,1]$, $r_{A} : \mathcal{S} \times \mathcal{V} \mapsto [0,1]$, $\pi_{B} : \mathcal{S} \times \mathcal{V} \mapsto \mathcal{A}$ and $P_{B} : \mathcal{S} \times \mathcal{V} \times \mathcal{A} \mapsto [0,1]$ where $\mathcal{S}, \mathcal{A}$ and $\mathcal{V}$ are respectively the sets of states, actions and messages.

[1] not to be confused with (Oh et al., 2018) which is an off-policy actor-critic algorithm promoting exploration in single-agent RL.

5.3.2 Practical Algorithm

ABIG iteratively structures the interactions between a builder-architect pair into interaction frames. Each iteration starts with a *modeling frame* during which the architect learns a model of the builder. Directly after, during the *guiding frame*, the architect leverages this model to produce messages that guide the builder. On its side, the builder stores the guiding interactions to train and refine its policy π_B. The interaction frames are described below. The algorithm is illustrated in Fig. 5.3 and the pseudo-code is reported in Algorithm 3.

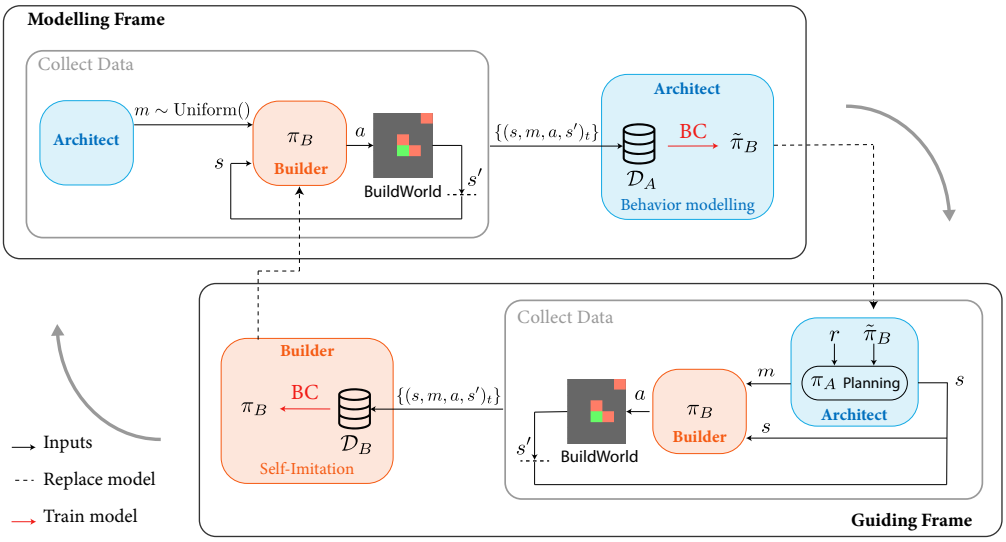


Figure 5.3: **Architect-Builder Iterated Guiding.** Agents iteratively interact through the modeling and guiding frames. In each frame, one agent collects data and improves its policy while the other agent's behavior is fixed.

Modeling Frame. The architect records a data-set of interactions $\mathcal{D}_A \triangleq \{(s, m, a, s')_t\}$ by sending random messages m to the builder and observing its reaction. After collecting enough interactions, the architect learns a model of the builder $\tilde{\pi}_B$ using BC (see Sec. 2.2).

Guiding Frame. During the guiding frame, the architect observes the environment states s and produces messages so as to maximize its return (see Eq. 5.3). The policy of the architect is a MCTS (see Sec. 2.1) that searches for the best message by simulating the reaction of the builder using $\tilde{a} \sim \tilde{\pi}_B(\cdot|m, s)$ alongside the dynamics and reward models. During this frame, the builder stores the interactions in a buffer $\mathcal{D}_B \triangleq \{(s, m, a, s')_t\}$. At the end of the guiding frame, the builder self-imitates by updating its policy π_B with BC on $\mathcal{D}_B$.

Practical Considerations. All models are parametrized by two-hidden layer 126-units feedforward ReLu networks. BC minimizes the cross-entropy loss with Adam optimizer (Kingma & Ba, 2015). Networks are re-initialized before each BC training. The architect's MCTS uses Upper-Confidence bound for Trees and relies on heuristics rather than Monte-Carlo rollouts to estimate the value of states. For more details about training, MCTS and hyper-parameters please see Suppl. Section B.3.

Algorithm 3: Architect-Builder Iterated Guiding (ABIG)

Require: randomly initialized builder policy π_B, reward function r, transition function P_E, BC algorithm, MCTS algorithm
for i in range($N_{iterations}$) **do**
  MODELLING FRAME:
    for e in range($N_{collect}/2$) **do**
      Architect populates $\mathcal{D}_A$ using $m \sim$ Uniform() and observing $a \sim \pi_B(\cdot|s, m)$
    end for
    Architect learns $\tilde{\pi}_B(a|s, m)$ on $\mathcal{D}_A$ with BC
    Architect sets $\pi_A(m|s) \triangleq$ MCTS($r, \tilde{\pi}_B, P_E$)
    Architect flushes $\mathcal{D}_A$
  GUIDING FRAME:
    for e in range($N_{collect}/2$) **do**
      Builder populates $\mathcal{D}_B$ using π_B while guided by Architect, i.e. $m \sim \pi_A(\cdot|s)$
    end for
    Builder learns $\pi_B(a|s, m)$ on $\mathcal{D}_B$ with BC
    Builder flushes $\mathcal{D}_B$
end for
Architect runs one last Modelling Frame
Result: π_A, π_B

The resulting method (ABIG) is general and can handle a variety of tasks while not restricting the kind of communication protocol that can emerge. Indeed, it only relies on a few high-level priors, namely, the architect's access to environment models, shared intent and interaction frames.

Control Settings. In addition to ABIG we also investigate two control settings: ABIG *-no-intent* – the builder interacts with an architect that disregards the goal and therefore sends random messages during training. At evaluation, the architect has access to the exact model of the builder ($\tilde{\pi}_B = \pi_B$) and leverages it to guide it towards the evaluation goal (the architect no longer disregards the goal). And *random* – the builder takes random actions. The comparison between ABIG and ABIG-no-intent measures the impact of doing self-imitation on guiding versus on non-guiding trajectories. The random baseline is used to provide a performance lower bound that indicates the task's difficulty.

5.3.3 Understanding the Learning Dynamics

Intuitive Explanation

Architect-Builder Iterated Guiding relies on two steps. First, the architect selects *favorable* messages, i.e. messages that maximize the likelihood of the builder picking optimal actions with respect to the architect's reward. Then, the builder does self-imitation and reinforces the guided behavior by maximizing the likelihood of the corresponding messages-actions sequence under its policy. The message-to-action associations (or preferences) are encoded in the builder's policy $\pi_B(a|s, m)$. Maximum likelihood assumes that actions are initially equiprobable for a given message. Therefore, actions under a message that is not present in the data-set ($\mathcal{D}_B$) remains so. In other words, if the builder never observes a message, it assumes that this message is equally associated with all the

possible actions. This enables the builder to *forget* past message-to-action associations that are not used – and thus not reinforced – by the architect. In practice, initial uniform likelihood is ensured by resetting the builder's policy network before each self-imitation. The architect can leverage the forget mechanism to erase unfavorable associations until a favorable one emerges. Such favorable associations can then be reinforced by the architect-builder pair until it is made deterministic. The *reinforcement* process of favorable associations is also enabled by the self-imitation phase. Indeed, for a given message m, the self-imitation objective for π on a data-set $\mathcal{D}$ collected using π is:

$$J(m,\pi) = -\sum_{a\sim\mathcal{D}} \log \pi(a|m) \approx \mathbb{E}_{a\sim\pi(\cdot|m)}[-\log \pi(a|m)] \approx H[\pi(\cdot|m)] \tag{5.5}$$

where H stands for the entropy of a distribution. Therefore, maximizing the likelihood, in this case, results in minimizing the entropy of $\pi(\cdot|m)$ and thus reinforces the associations between messages and actions. Using these mechanisms the architect can adjust the policy of the builder until it becomes *controllable*, i.e. deterministic (strong preferences over actions for a given message) and flexible (varied preferences across messages). Conversely, in the case of ABIG-no-intent, the architect does not guide the builder and simply sends messages at random. Favorable and unfavorable messages are thus sampled alike which prevents the forgetting mechanism to undo unfavorable message-to-action associations. Consequently, in that case, self-imitation tends to simply reinforce the initial builder's preferences over actions making the controllability of the builder policy depend heavily on the initial preferences.

ABIG with a Toy Problem

To illustrate the learning mechanisms of ABIG we propose to look at the simplest instantiation of the Architect-Builder Problem: there is one state (thus it can be ignored), two messages m_1 and m_2 and two possible actions a_1 and a_2. If the builder chooses a_1 it is a loss ($r(a_1) = -1$) but choosing a_2 results in a win ($r(a_2) = 1$). Fig. 5.4 displays several iterations of ABIG on this problem when the initial builder's policy is unfavorable (a_1 is more likely than a_2 for all the messages). During each iteration, the architect selects messages in order to maximize the likelihood of the builder picking action a_2 and then the builder does self-Imitation Learning by maximizing the likelihood of the corresponding messages-actions sequence under its policy. Fig. 5.4 shows that this process leads to forgetting unfavorable associations until a favorable association emerges and can be reinforced. On the other hand, for ABIG-no-intent in Fig. 5.5, favorable and unfavorable messages are sampled alike which prevents the forgetting mechanism to undo unfavorable message-to-action associations. Consequently, initial preferences are reinforced.

To further assess how the architect's message choices impact the performance of a self-imitating builder, we compare the distribution of the builder's preferred actions obtained after using ABIG and ABIG-no-intent. We consider three different initial conditions (favorable, unfavorable, intermediate) that are each ran to convergence (meaning that the policy does not change anymore across iterations) for 100 different seeds.

Fig. 5.6 displays the resulting distributions of preferred – i.e. most likely – action for each message. When applying ABIG on the toy problem, the pair always reaches a success

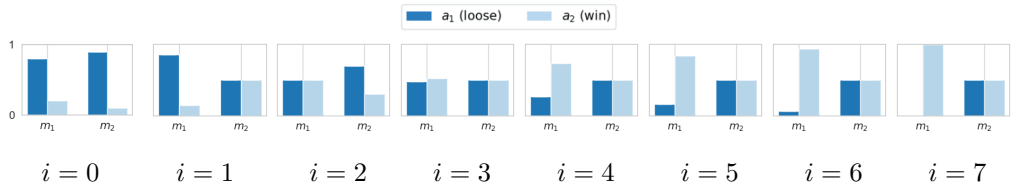


Figure 5.4: ABIG-driven evolution of message-conditioned action probabilities in the toy problem. Initial conditions are unfavorable since a_1 is more likely than a_2 for both messages. ($i = 0$) Given the initial conditions, the architect only sends message m_1 since it is the most likely to result in action a_2. ($i = 1$) the builder guiding data only consisted of m_1 message therefore it cannot learn a preference over actions for m_2 and both actions are equally likely under m_2. The architect now only sends message m_2 since it is more likely than m_1 at triggering a_2. ($i = 2$) Unfortunately, the sampling of m_1 resulted in the builder doing more a_1 than a_2 during the guiding frame and the builder thus associates m_2 with a_1. The architect tries its luck again but now with m_1. ($i = 3$) Eventually, the sampling results in more a_2 actions being sampled in the guiding data and the builder now associates m_1 to a_2. ($i = 4$) and ($i = 5$) The architect can now keep on sending m_1 messages to reinforce this association.

Figure 5.5: ABIG-no-intent driven evolution of message-conditioned action probabilities for a simple problem where builder must learn to produce action a_2. Initial conditions are unfavorable since a_1 is more likely than a_2 for both messages. Without an architect's guiding messages during training, a self-imitating builder reinforces the action preferences of the initial conditions and fails (even when evaluated alongside a knowledgeable architect as both messages can only yield a_1).

rate of 100/100 no matter the initial condition. We also observe that, at convergence, the builder never prefers action a_1, yet when an action is preferred for a given message, the other message yields no preference over action ($p(a_1|m) = p(a_2|m)$). This is due to the forgetting mechanism. The results when applying ABIG-no-intent on the toy problem are much more dependent on the initial condition. In the unfavorable scenario, ABIG-no-intent fails heavily with only 3 seeds succeeding over the 100 experiments. This is due to the fact that, in absence of message guidance from the architect, the builder has a high chance to continually reinforce the association between the two messages and a_1, therefore losing. However, in rare cases, the builder can inverse the initial message-conditioned probabilities by 'luckily' sampling more often a_2 when receiving m_1 and win. This only happened 3 times over the 100 seeds. Finally, when initial conditions are more favorable, the self-imitation steps reinforce the association between the messages and a_2 which makes the builder prefer a_2 for at least one message and enables high success rates (100/100 for favorable and 98/100 for intermediate).

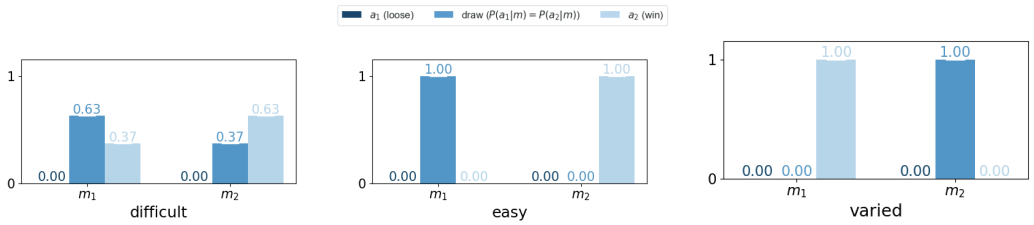
Interestingly, the emergent learning mechanisms discussed here are reminiscent of the amplification and self-enforcement of random fluctuations in naming games (Steels, 1995a). In language games, however, the self-organization of vocabularies is driven by

Unfavorable	Favorable	Intermediate

(a) **Initial probabilities**

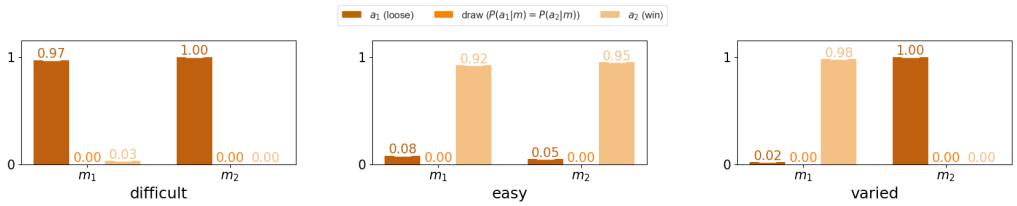
$P(a_1\|m_1) = 0.8,\ P(a_2\|m_1) = 0.2$ $P(a_1\|m_2) = 0.9,\ P(a_2\|m_2) = 0.1$	$P(a_1\|m_1) = 0.2,\ P(a_2\|m_1) = 0.8$ $P(a_1\|m_2) = 0.1,\ P(a_2\|m_2) = 0.9$	$P(a_1\|m_1) = 0.9,\ P(a_2\|m_1) = 0.1$ $P(a_1\|m_2) = 0.1,\ P(a_2\|m_2) = 0.9$

(b) **ABIG:** Distributions of final preferred action for each message calculated over 100 seeds



Success Rate = 100/100 | Success Rate = 100/100 | Success Rate = 100/100

(c) **ABIG-no-intent:** Distributions of final preferred action for each message calculated over 100 seeds



Success Rate = 3/100 | Success Rate = 100/100 | Success Rate = 98/100

Figure 5.6: **Toy experiment analysis** (a) Initial conditions: initial probability for each action a given a message m; distributions of final builder's preferred actions for each message after applying (b) ABIG and (c) ABIG-no-intent on the toy problem; distributions are calculated over 100 seeds.

each agent maximizing its communicative success whereas in our case the builder has no external learning signal and simply self-imitates.

5.3.4 Related Work

This work is inspired by experimental semiotics (Galantucci & Garrod, 2011) and in particular (Vollmer et al., 2014) that studied the CoCo game with human subjects as a key step towards understanding the underlying mechanisms of the emergence of communication. Here we take a complementary approach by defining and investigating solutions to the ABP, a general formulation of the CoCo game where both agents are AIs.

Recent MARL work (Lowe et al., 2017; Woodward et al., 2020; Roy et al., 2020; Ndousse et al., 2021), investigate how RL agents trained in the presence of other agents leverage the behaviors they observe to improve learning. In these settings, the other agents are used to build useful representation or gain information but the main learning signal of every agent remains a ground truth reward.

Feudal Learning (Dayan & Hinton, 1992; Kulkarni et al., 2016; Vezhnevets et al., 2017; Nachum et al., 2018; Ahilan & Dayan, 2019) investigate a setting where a manager sets the rewards of workers to maximize its own return. In this Hierarchical setting,

the manager interacts by directly tweaking the workers' learning signal. This would be unfeasible for physically distinct agents, hence those methods are restricted to single-agent learning. On the other hand, ABP considers separate agents, that must hence communicate by influencing each other's observations instead of rewards signals.

IRL has been investigated for HRI when it is challenging to specify a reward function. Instead of defining rewards, IRL rely on expert demonstrations. Hadfield-Menell et al. (2016) argue that learning from expert demonstrations is not always optimal and investigate how to produce instructive demonstrations to best teach an apprentice. Crucially, the expert is aware of the mechanisms by which the apprentice learns, namely RL on top of IRL. This allows the expert to assess how its demonstrations influence the apprentice policy, effectively reducing the problem to a single agent POMDP. In our case, however, the architect and the builder do not share the same action space which prevents the architect from producing demonstrations. In addition, the architect ignores the builder's learning process which makes the simplification to a single-agent teacher problem impossible.

In essence, the ABP is closest to works tackling the calibration-free BCI control problem (Grizou et al., 2014; Xie et al., 2021). Yet, these works both consider that the architect sends messages after the builder's actions and thus enforce that the feedback conveys a reward. Crucially, the architect does not learn and communicates with a fixed mapping between feedback and pre-defined meanings ("correct" vs. "wrong"). Those meanings are known to the builder and it simply has to learn the mapping between feedback and meaning. In our case, however, the architect communicates before the builder's action and thus rather gives instructions than feedback. Additionally, the builder has no a priori knowledge of the set of possible meanings and the architect adapts those to the builder's reaction. Finally, Grizou et al. (2013) handles both feedback and instruction communications but relies on known task distribution and a set of possible meanings. In terms of motivations, previous works are interested in one robot figuring out a fixed communication protocol while we train two agents to collectively emerge one.

Our BuildWorld resembles GridLU proposed by Bahdanau et al. (2019b) to analyze reward modeling in language-conditioned learning. However, their setting is fundamentally different to ours as it investigates single agent goal-conditioned IL where goals are predefined episodic linguistic instructions labelling expert demonstrations. Nguyen et al. (2021) alleviate the need for expert demonstrations by introducing an interactive teacher that provides descriptions of the learning agent's trajectories. In this HRI setting, the teacher still follows a fixed pre-defined communication protocol known by the learner: messages are activity descriptions. Our ABP formulation relates to the Minecraft Collaborative Building Task (Narayan-Chen et al., 2019) and the IGLU competition (Kiseleva et al., 2021); however, they do not consider emergent communication. Rather, they focus on generating architect utterances by leveraging a human-human dialogues corpus to learn pre-established meanings expressed in natural language. Conversely, in ABP both agents learn and must evolve the meanings of messages while solving the task without relying on any form of demonstration.

5.4 Experiments

In the following sections, success rates (sometimes referred as scores) are averaged over 10 random seeds and error bars are ±2SEM with SEM the Standard Error of the Mean. If not stated otherwise, the grid size is (5×6), contains three blocks ($N_b = 3$) and the vocabulary size is $|\mathcal{V}| = 18$.

5.4.1 ABIG's Learning Performances

We apply ABIG to the four learning tasks of BuildWorld and compare it with the two control settings: ABIG-no-intent (no guiding during training) and random (builder takes random actions). Fig. 5.7 reports the mean success rate on the four tasks defined in Sec. 5.2. First, we observe that ABIG significantly outperforms the control conditions on all tasks. Second, we notice that on the simpler 'grasp' task ABIG-no-intent achieves a satisfactory mean score of 0.77±0.03. This is consistent with the learning dynamic analysis provided in 5.3.3 that shows that, in favorable settings, a self-imitating builder can develop a reasonably controllable policy (defined in Sec. 5.3.3) even if it learns on non-guiding trajectories. Nevertheless, when the tasks get more complicated and involve placing objects or drawing lines, the performances of ABIG-no-intent drop significantly whereas ABIG continues to achieve high success rates (> 0.8). This demonstrates that ABIG enables a builder-architect pair to successfully agree on a communication protocol that makes the builder's policy controllable and enables the architect to efficiently guide it.

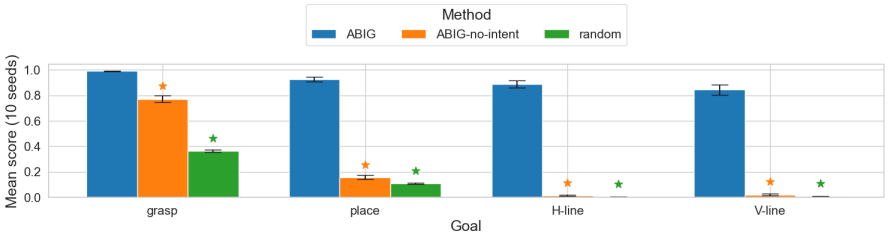


Figure 5.7: Methods performances (stars indicate significance with respect to ABIG model according to Welch's t-test with null hypothesis $\mu_1 = \mu_2$, at level $\alpha = 0.05$). ABIG outperforms control baselines on all goals.

5.4.2 ABIG's Transfer Performances

Building upon previous results, we propose to study whether a learned communication protocol can transfer to new tasks. The architect-builder pairs are trained on a single task and then evaluated without retraining on the four tasks. In addition, we include 'all-goals': a control setting in which the builder learns a single policy by being guided on all four goals during training. Fig. 5.8 shows that, on all training tasks except 'grasp', ABIG enables a transfer performance above 0.65 on all testing tasks. Notably, training on 'place' results in a robust communication protocol that can be used to solve the other tasks with a success rate above 0.85, being effectively equivalent as training on 'all-goals'

directly. This might be explained by the fact that placing blocks at specified locations is an atomic operation required to build lines.

Figure 5.8: ABIG transfer performances without retraining depending on the training goal. ABIG agents learn a communication protocol that transfers to new tasks. Highest performances reached when training on 'place'.

Challenging ABIG's transfer abilities. Motivated by ABIG's transfer performances, we propose to train it on the 'place' task in a bigger grid (6×6) with $N_b = 6$ and $|\mathcal{V}| = 72$. Then, without retraining, we evaluate it on the '6-block-shapes' task[2] that consists in constructing the shapes given in Fig. 5.9. The training performance on 'place' is 0.96 ± 0.02 and the transfer performance on the '6-block-shapes' is 0.85 ± 0.03. This further demonstrates ABIG's ability to derive robust communication protocols that can solve more challenging unseen tasks.

Figure 5.9: 6-block-shapes that ABIG can construct in transfer mode when trained on the 'place' task.

5.4.3 Proof of Emerging Language

In this paragraph, we propose to thoroughly study the evolution of the builder's policy in order to provide a deeper analysis of ABIG. Our analysis principally relies on mutual information measures that we define below.

Metric definition. We define three metrics that characterize the builder's behavior. We compute these metrics on a constant *Measurement Set* $\mathcal{M}$ made of 6000 randomly sampled states, for each of these states we sample all the possible messages $m \sim \text{Uniform}(\mathcal{V})$ where $\mathcal{V}$ is the set of possible messages. Therefore, $|\mathcal{M}| = 6000 \times |\mathcal{V}|$. The set of possible actions is $\mathcal{A}$ and we denote by δ the indicator function.

[2]For rollouts see https://sites.google.com/view/architect-builder-problem/

We also define the following distributions:

$$
\begin{aligned}
p_{S}(s) &\triangleq \frac{1}{|\mathcal{M}|} \sum_{s' \in \mathcal{M}} \delta(s' == s) \\
p_{M}(m) &\triangleq P(m|s) = \frac{1}{|\mathcal{V}|} \\
p_{SM}(s,m) &\triangleq p_{S}(s) P(m|s) = p_{S}(s) p_{M}(m) \\
p_{SMA}(s,m,a) &\triangleq p_{SM}(s,m) P(a|s,m) = p_{SM}(s,m) \pi_{B}(a|s,m) \\
p_{A}(a) &\triangleq \sum_{(s,m) \in \mathcal{M}} p_{SMA}(s,m,a) \\
p_{MA}(m,a) &\triangleq \sum_{s \in \mathcal{M}} p_{SMA}(s,m,a) \\
p_{SA}(s,a) &\triangleq \sum_{m \in \mathcal{M}} p_{SMA}(s,m,a)
\end{aligned}
$$

From this we can define the monitoring metrics:

- *Mean Entropy:*

$$\bar{H}(\pi) = \frac{1}{|\mathcal{M}|} \sum_{(s,m) \in \mathcal{M}} \left[- \sum_{a \in \mathcal{A}} \pi(a|s,m) \log \pi(a|s,m) \right]$$

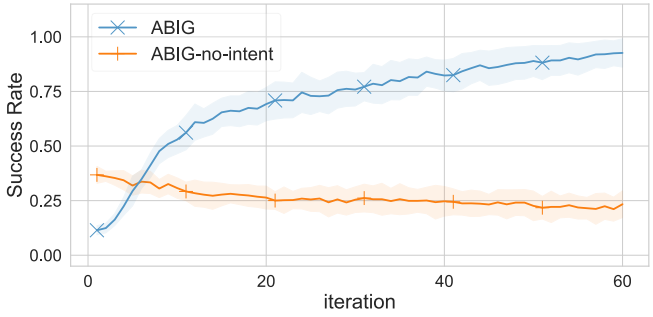
- *Mutual Information between messages and actions*

$$I_m = \sum_{m \in \mathcal{V}} \sum_{a \in \mathcal{A}} p_{MA}(m,a) \log \frac{p_{MA}(m,a)}{p_{A}(a) p_{M}(m)}$$

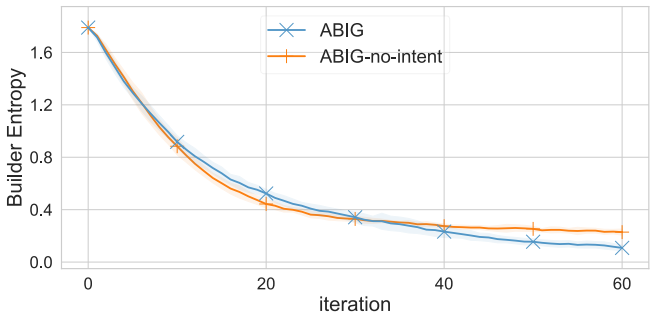
- *Mutual Information between states and actions*

$$I_s = \sum_{s \in \mathcal{M}} \sum_{a \in \mathcal{A}} p_{SA}(s,a) \log \frac{p_{SA}(s,a)}{p_{A}(a) p_{S}(s)}$$

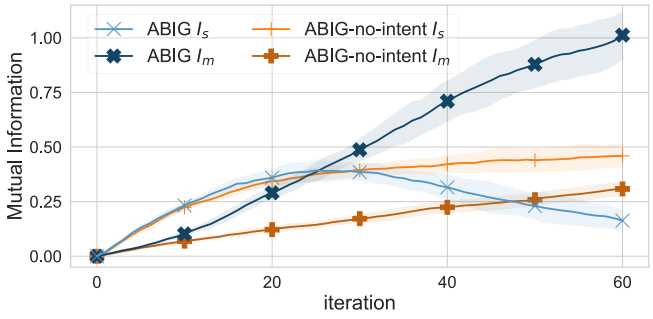
Analysis. Fig. 5.10 displays the evolution of these metrics after each iteration as well as the evolution of the success rate (a). As indicated by Eq. (5.5), doing self-imitation learning results in a decay of the mean entropy (b). This decay is similar for ABIG and ABIG-no-intent. The most interesting result is provided by the evolution of the mutual information (c). For ABIG-no-intent, we see that I_s and I_m slowly increase with $I_s > I_m$ over all iterations. This indicates that the builder policy $\pi_B(a|s,m)$ relies more on states than on messages to compute the actions. In this scenario the builder, therefore, tends to ignore messages. On the other hand, I_s and I_m evolve differently for ABIG. Both metrics first increase with $I_s > I_m$ until they cross around iteration 25. Then I_s starts decreasing and I_m grows. This shows that ABIG results in a builder policy that strongly selects actions based on the messages it receives which is a desirable feature of emergent communication.



(a) Evolution of the success rate



(b) Evolution of the builder policy mean entropy $\bar{H}_{\pi_B}$



(c) Evolution of the mutual information I_s and I_m

Figure 5.10: Comparison of the evolution of builder policy properties when applying ABIG and ABIG-no-intent on the 'place' task in BuildWorld. (a) ABIG enables much higher performance that ABIG-no-intent. (b) Both methods use self-imitation and thus reduce the entropy of the policy. (c) ABIG promotes the mutual information between messages and action which indicates successful communication protocols.

5.4.4 Additional Baselines

We define two extra baselines:

- Stochastic: where the builder policy is a fixed softmax policy parameterized by a randomly initialized network;
- Deterministic: where the builder policy is a fixed argmax policy parameterized by a randomly initialized network.

In the performances reported in Fig. 5.11, the architect has direct access to the exact policy of the builder ($\tilde{\pi}_B = \pi_B$) and uses it to plan and guide the builder during evaluation. We observe that the stochastic condition exhibits similar performances as the random builder. This indicates that, even if the architect tries to guide the builder, the stochastic policy is not controllable and performances are not improved. Finally, we would expect a deterministic policy to be more easily controllable by the architect. Yet, as pointed out in Fig. 5.11, the initial deterministic policies lack flexibility and fail. This shows that the builder must iteratively evolve its policy in order to make it controllable.

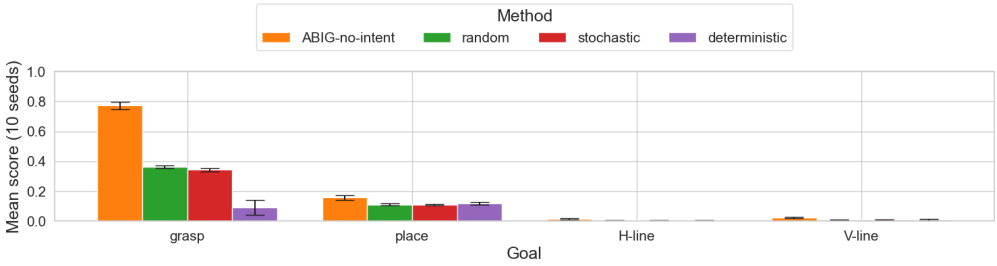


Figure 5.11: Baseline performance depending on the goal: stochastic policy behaves on par with random builder. Self-imitation with ABIG-no-intent remains the most controllable baseline.

5.4.5 Impact of Vocabulary Size

We finally investigate the impact of vocabulary size on ABIG communicative performance in Fig. 5.12. The bigger the vocabulary size, the better the performances suggesting that with more messages available, the architect can more efficiently refer to the desired action.

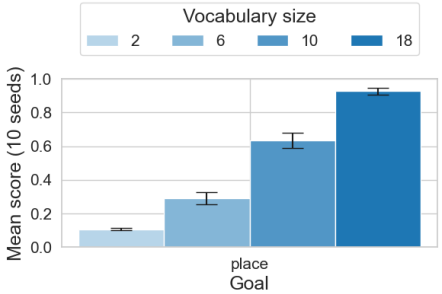


Figure 5.12: Influence of the Vocabulary size for ABIG on the 'place' task. Performance increases with the vocabulary size.

5.5 Discussion and Future Work

This work formalizes the ABP as an interactive setting where learning must occur without explicit reinforcement, demonstrations, or a shared language. To tackle ABP, we propose ABIG: an algorithm that learns to guide and to be guided. ABIG is based only on two high-level priors to communication emergence (shared intent and interactions frames). ABP's general formulation allows us to formally enforce those priors during learning. We study their influence through ablation studies, highlighting the importance of shared intent achieved by doing self-imitation on guiding trajectories. When performed in interaction frames, this mechanism enables agents to evolve a communication protocol that allows them to solve all the tasks defined in BuildWorld. More impressively, we find that communication protocols derived on a simple task can be used to solve harder, never-seen goals.

Our approach has several limitations which open up different opportunities for further work. First, ABIG trains agents in a stationary configuration which implies doing several interaction frames. Each interaction frame involves collecting numerous transitions. Thus, ABIG is not data efficient. A challenging avenue would be to relax this stationarity constraint and have agents learn from buffers containing non-stationary data with obsolete agent behaviors. Second, the builder remains dependent on the architect's messages even at convergence. Using a Vygotskian approach, the builder could internalize the guidance from the architect to become autonomous in the task. This could, for instance, be achieved by having the builder learn a model of the architect's message policy once the communication protocol has converged.

Because we present the first step towards interactive agents that learn in the ABP, our method uses simple tools (feed-forward networks and self-imitation learning). It is however important to note that our proposed formulation of the ABP can support many different research directions. Experimenting with agents' models could allow for the investigation of other forms of communication. One could, for instance, include memory mechanisms in the models of agents in order to facilitate the emergence of retrospective feedback, a form of emergent communication observed in (Vollmer et al., 2014). ABP is also compatible with low-frequency feedback. As a further experiment in this direction, one could penalize the architect for sending messages and assess whether a pair can converge to higher-level meanings. Messages could also be composed of several tokens in order to allow for the emergence of compositionality. Finally, our proposed framework can serve as a testbed to study the fundamental mechanisms of emergent communication by investigating the impact of high level communication priors from experimental semiotics.

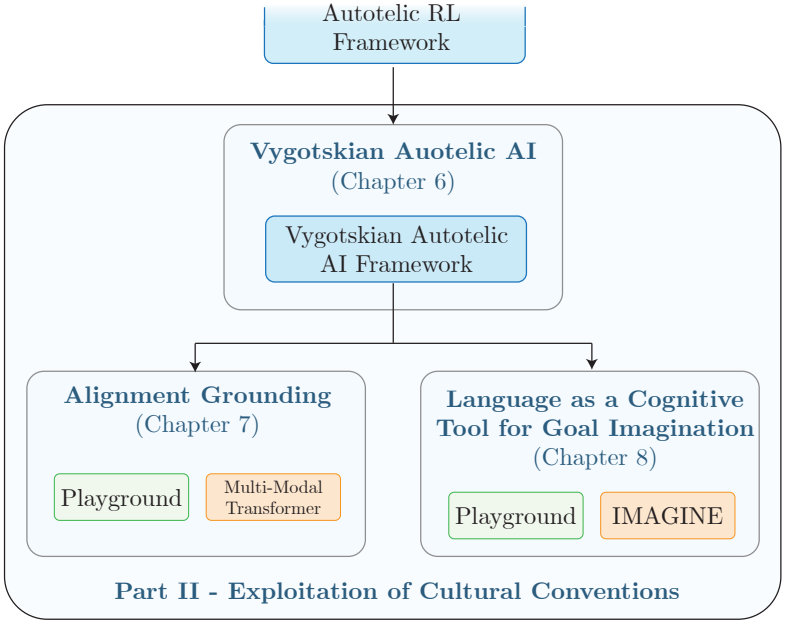
Part Summary

The first part of this manuscript built on the language formation framework introduced in Sec. 3.2 to present two complementary experimental contributions. In chapter 4 we presented the graphical referential game, an ecological extension of the standard language games to a context in which agents must learn to communicate via a sensory-motor graphical apparatus. To tackle the graphical referential game, we proposed CURVES an algorithm enabling agents to train contrastive representations of visual inputs and graphical utterances and to achieve successful communication.

In chapter 5, we introduced a new paradigm to investigate the emergence of goal-directed communication. More specifically, we presented the Architect-Builder Problem, a setting inspired by the experimental semiotics study of Vollmer et al. (2014) in which agents have asymmetries of information (only the architect knows the goal) and asymmetries of affordances (only the builder can act in the environment). To tackle the ABP, we introduced ABIG: architect builder iterated guiding. ABIG relies on the interaction frame and shared intent priors to enable agents to successfully communicate and solve several instantiations of the ABP.

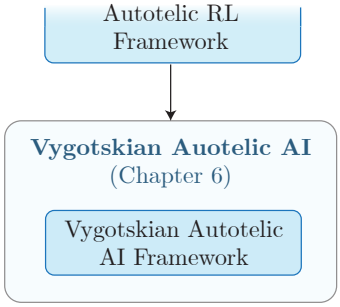
Part II

Exploitation of Cultural Conventions



Chapter 6

Vygotskian Autotelic AI



Contents

The second part of this research focuses on the exploitation of cultural conventions by artificial agents with the aim of building open-ended repertoires of skills and self-organizing their developmental trajectories. In the first chapter of part II, we present a new AI framework coined Vygotskian Autotelic AI (VAAI). This framework extends the autotelic RL framework presented in 3.3.1. It proposes to immerse artificial agents in rich socio-cultural worlds in order to convert the cultural conventions they take part in into cognitive tools enabling them to enhance their learning capabilities.

6.1 Motivations

The introduction of the VAAI is motivated by the serious limitations exhibited by recent autotelic approaches. The goal representations emerging from their intrinsically motivated experience with the physical world end up very concrete and mostly consist in

reaching target stimuli (e.g. matching their visual input with a particular target). This contrasts with the wide diversity and the abstraction of goals targeted by humans. In addition, the generated goals very often belong to the distribution of previously experienced effects, which drastically limits the ability of autotelic agents to represent *creative goals*, thus to explore and undergo an open-ended discovery process. Besides goal imagination, RL algorithms still lack human-like capacities in terms of generalization, skill composition, abstraction, or sample efficiency (Witty et al., 2021; Shanahan & Mitchell, 2022)

Just like Piaget's theory of child development inspired developmental AI (Dautenhahn & Billard, 1999) and is at the root of autotelic RL, VAAI draws inspiration from the famous developmental psychologist: Vygotsky. Humans are social beings; intrinsically motivated to interact and cooperate with their peers (Tomasello, 1999b; Tomasello et al., 2005; Brewer et al., 2014). For Vygotsky, linguistic social interactions such as descriptions, explanations, corrections, or play start as interpersonal processes before they are turned into *intrapersonal* cognitive processes through the process of *internalization* (Vygotsky, 1934) Following his vision, many psychologists (Berk, 1994; Lupyan, 2012; Gentner & Hoyos, 2017), linguists (Whorf, 1956; Rumelhart et al., 1986; Lakoff & Johnson, 2008), and philosophers (Hesse, 1988; Dennett, 1993; Clark, 1998; Carruthers, 2002) argued for the importance of socio-cultural interactions in the development of human intelligence.

In VAAI, we propose to include socio-cultural interactions in the learning environment of the agent. To do so, we propose to immerse them into rich socio-cultural worlds; to let them interact with us and with their peers in natural language; to let them internalize these interactions and mesh them with their cognitive development. Just like they do for humans, language and culture will help shape the agents' goal representations and generation mechanisms, thereby offering them the ability to generate more diverse and abstract goals; to imagine new goals beyond their past experience. Because they will develop at our contact, bathed in our cultures, they will learn about our cultural norms, values, customs, interests, and thought processes; all of which would be impossible to learn in social isolation. Just like humans, machines will use language to develop higher cognitive functions like abstraction, generalization, or imagination (Carruthers & Boucher, 1998; Gentner & Hoyos, 2017; Dove, 2018).

This chapter extends previous calls to leverage Vygotsky's insights for more socially-situated cognitive robotics (Dautenhahn & Billard, 1999; Zlatev, 2001; Lindblom & Ziemke, 2003; Mirolli & Parisi, 2011). Zlatev discussed interactions between social-situatedness and epigenetic development (Zlatev, 2001), Dautenhahn and Billard drew the parallel between AI and the Piagetian vs. Vygotskian views (Dautenhahn & Billard, 1999), while Mirolli and Parisi, as well as Cangelosi et al., reviewed the first successful auxiliary uses of language for decision-making (Mirolli & Parisi, 2011; Cangelosi et al., 2010a). In the last decade however, the AI community seems to have lost track of these insights. Today we update these arguments in the light of recent AI advances and reframe the Vygotskian perspective within the autotelic RL framework.

The next section sets the background and discusses the interaction between language and thought in humans by building on the work of psychologists and philosophers (Sec. 6.2). Then the following section formally presents the components of Vygotskian autotelic agents. The last section finally reframes recent contributions at the intersec-

tion of RL and language in the light of the VAAI framework detailing 1) the ability to exploit the information contained in linguistic structure and content (syntax, vocabulary, narratives) to support the development of cognitive functions (Sec. 6.4.1); 2) the ability to internalize linguistic interactions within the agent to power its future autonomy and integration into the socio-cultural world (Sec. 6.4.2).

6.2 Language and Thought in Humans, a Vygotskian Perspective

Our ability to generate new ideas is the source of our incredible success in the animal kingdom. But this ability did not appear with the first *homo sapiens* 130,000 years ago. Indeed, the oldest imaginative artifacts such as figurative arts, elaborate burials, or the first dwellings only date back to 70,000 years ago (Harari, 2014; Vyshedskiy, 2019). This is thought to coincide with the apparition of *recursive language* (Goldberg, 1999; Vyshedskiy, 2019; Hoffmann, 2020). Which of these appeared first? Creativity or recursive language? Or did they mutually bootstrap?

Extreme views on the topic either characterize language as a pure communicative device to convey our inner thoughts (strong communicative thesis) (Chomsky, 1957b; Fodor, 1975) or, on the other hand, argue that only language can be the vehicle of our thoughts (strong cognitive thesis) (Wittgenstein, 1953; McDowell, 1996). As often with binary oppositions, the truth seems to lie in between. Animal and preverbal infants demonstrate complex cognition (Sperber et al., 1995; Allen & Bekoff, 1999) but language does impact the way we perceive (Waxman & Markow, 1995; Yoshida & Smith, 2003), represent concepts (Lakoff & Johnson, 2008), conduct compositional and relational thinking (Gentner & Loewenstein, 2002; Gentner & Hoyos, 2017; Vyshedskiy, 2019) etc. Thus, language seems to be at least *required* to develop some of our cognitive processes (requirement thesis), and might still be the vehicle of *some* of our thoughts (constitutive thesis) (Carruthers & Boucher, 1998). Interested readers can find a thorough overview of this debate in *Language and Thought* by Carruthers & Boucher (1998)

If language is required to develop some of our higher cognitive functions, then autotelic artificial agents should use it as well. But how does that work? What is so special about language? Let us start with *words*, which some called *invitations to form categories* (Waxman & Markow, 1995). Hearing the same word in a variety of contexts invites humans to compare situations, find similarities, differences and build symbolic representations of agents, object and their attributes. With words, the continuous world can be simplified and structured into mental entities at various levels of abstraction.

The recursivity and partial compositionality of language allow us to readily understand the meaning of sentences we never heard before by generalizing from known words and syntactic structures. On the flip side, it also supports *linguistic productivity* (Chomsky, 1957b) the ability to generate new sentences — thus new ideas — in an open-ended way. Relational structures such as comparisons and metaphors facilitate our relational thinking (Gentner & Loewenstein, 2002; Gentner & Hoyos, 2017), condition our ability to compose mental images (Vyshedskiy, 2019), and support our understanding of abstract concepts such as emotions, politics or scientific theories (Hesse, 1988; Lakoff & Johnson, 2008).

Finally, language is a cultural artefact inherited from previous generations and shared with others. It supports our cultural evolution and allows humans to efficiently transfer knowledge and practices across people and generations (Henrich & McElreath, 2003; Morgan et al., 2015; Chopra et al., 2019) — a process known as the *cultural rachet* (Tomasello, 1999b). Through shared cultural artefacts such as narratives, we learn to share common values, customs, and social norms, we learn how to navigate the world, what to attend to, how to think, and what to expect from others (Bruner, 1990). This cultural knowledge is readily accessible to children as they enter societies via social interactions and formal education. Learning language further extends access to cultural artifacts such as books, movies, or the Internet. These act as a thousand virtual social partners to learn from.

We now understand why language is so special. Let us focus on how it can shape cognitive development in humans and machines. Dennett, a proponent of the requirement thesis, suggests that linguistic exposition alone can lead to a fundamental cognitive reorganization of the human brain (Dennett, 1993). He compares it to the installation of a serial virtual machine on humans' massively parallel processing brains. As a result, a slight change in our computational hardware (e.g. compared to our primate relatives) could open the possibility for any cognitive software reprogramming driven by language, in turn triggering the learning and cultural evolution of higher cognitive capacities.

Carruthers, a proponent of the constitutive thesis, suggests that language may have evolved as a separate module to exchange inner representations with our peers (naive physics, theory of mind, etc). This would require connections between linguistic and non-linguistic modules to allow conversions between inner representations and linguistic inputs/outputs. In a similar way that humans can trigger imagined visual representations via top-down connections in their visual cortex, top-down activations of the linguistic module would create *inner speech*. This hallucinated speech, when broadcast to other modules, would implement *thinking in language* (Carruthers, 1998). Clark advances yet another possibility, the *supra-communicative view*. Here, language does not transform the way the brain makes computations and is not the vehicle of thoughts. Instead, language complements our standard computation activities by "*re-shaping the computational spaces*," turning problems that would be out of reach into problems our pattern-matching brains can solve (Clark, 1998). In that sense, language is a *cognitive tool* that enhances our cognitive abilities without altering them per se.

Vygotsky's theory brings a complementary argument to this debate. Caretakers naturally scaffold the learning experiences of children, tailoring them to their current objectives and capacities. Through encouragement, attention guidance, explanations or plan suggestions, they provide cognitive aids to children in the form of interpersonal social processes (Vygotsky, 1934). In this *zone of proximal development*, as Vygotsky coined it, children can benefit from these social interactions to achieve more than they could alone. In these moments, children *internalize* linguistic and social aids and progressively turn these interpersonal processes into intrapersonal *psychological tools* (Vygotsky, 1934). This essentially consists in building internal models of social partners such that learners can self-generate contextual guidance in the absence of an external one. Social speech is internalized into private speech (an outer speech of children for themselves), which, as it develops, becomes more goal-oriented and provides cognitive aids of the type caretakers would provide (Vygotsky, 1934; Berk, 1994). Progressively, it becomes more efficient

and abbreviated, less vocalized, until it is entirely internalized by the child and becomes *inner speech*.

This section showed why language is so important and might just be required for the development of our highest cognitive functions. If we want machines to show more human-like open-ended skill discovery processes, we might need to immerse them into rich socio-cultural worlds from the very beginning — just like we do with children — and equip them with tools to benefit from them. The next section leverages these observations to outline the components of vyogtskian autotelic agents.

6.3 Vygotskian Autotelic Artificial Intelligence (VAAI)

Fig. 6.1 illustrates the key elements of VAAI. Once immersed in socio-cultural worlds (a), vygotksian autotelic agents first need to ground the meaning of socio-cultural interactions in their physical experience of the world. To do so they need to extract the information contained within linguistic structures/contents in order to map it to their sensory-motor modalities. This grounding phase is achievable by training extractive language-conditioned models as illustrated in Fig. 6.1(b). For instance, language and observations can be directly mapped to actions via a language-conditioned policy, or to rewards converting language predicates into a learning signal for RL agents. When exposed to language, agents will reorganize their internal representations for better abstraction, generalization, and better alignment with human values, norms, and customs (Dennett's thesis). We will examine recent works that focus on training extractive models in Sec. 6.4.1 and we will present a transformer-based neural architecture for grounding the meaning of linguistic descriptions of behavior in chapter 7.

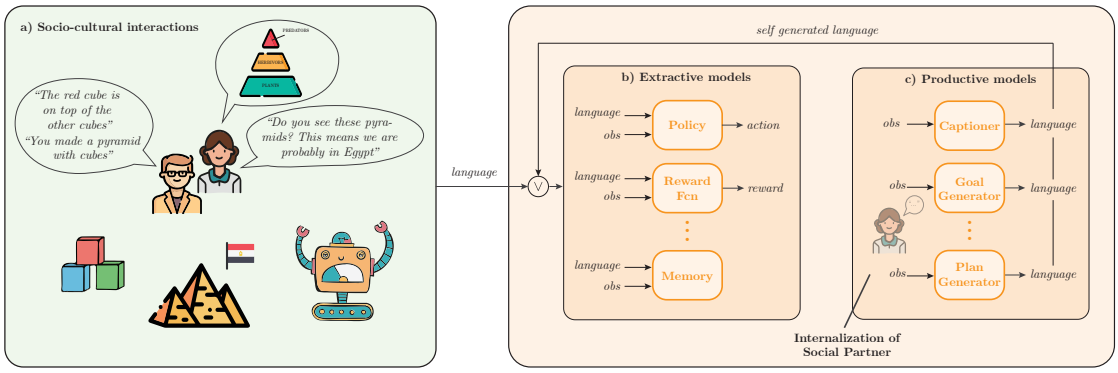


Figure 6.1: The three components of Vygotskian autotelic agents: socio-cultural interactions, linguistic extraction and internalized linguistic production. Vygotskian autotelic agents are immersed into rich socio-cultural worlds where they experience a variety of linguistic feedback including descriptions, explanations, or metaphors (a). They can exploit information from linguistic structures and content by conditioning their internal modules on this feedback (b, extractive models). Finally, they learn to internalize social interactions by training productive models of language to generate feedback similar to the one they receive from others (c, productive models). This offers agents the autonomy to build their own cognitive tools, bootstrapped by socio-cultural language.

Second, we need autotelic agents to internalize social interactive processes, i.e. to *model social partners within themselves* (Vygotsky's internalization). Social processes, turned into intrapersonal cognitive processes will orient the agent's focus, help it decompose tasks or imagine goals. This inner speech generation can serve as a common currency between other modules (e.g. perception, motor control, goal generation) in line with Carruther's view and will help agents project problems onto linguistic spaces where they might be easier to solve (Clark's view). As illustrated in Fig. 6.1(c), productive models take observations as input and output language. They can be used to generate plans or goal candidates in autotelic RL. We will review recent approaches leveraging productive models in Sec. 6.4.2. We will also present the IMAGINE agent, a Vygotskian autotelic agent, in chapter 8.

6.4 Recent Related Work

6.4.1 Exploiting Linguistic Structure and Content

In its vocabulary, syntax and narratives, language offers both powerful computational tools for thinking and important cultural knowledge about the world. According to Dennett's thesis, mere exposure to language can already help agents rewire their inner processes and develop new abilities. Recent advances in AI seem to support that idea.

Learning to abstract and generalize

Exposition to linguistic labels is known to facilitate category learning in humans (Waxman & Markow, 1995; Yoshida & Smith, 2003), but also in machines (Lupyan, 2005). As Mirolli and Parisi defend, the repeated occurrence of a linguistic label (*red* in their example) leads to the conflation of internal representations associated with that label (red things) which, in turn, facilitates further classifications based on the linguistic attributes (Mirolli & Parisi, 2011).

We see a similar effect in RL agents targeting linguistic goals. The exposure to aligned instructions and trajectories seems to reshape the internal representations of the agent contained within its action policy. The policy is a neural network-based function conditioned on the agent's instruction that maps the current state of the world to its next actions. By *internal representations*, we mean representations computed within the layers of the policy to facilitate the final decision-making. When repeatedly asked to grasp *red objects*, the policy learns to focus on objects' colors to facilitate action selection (Hill et al., 2020a). *Red* is an abstraction over a continuous space of colors. It is first cultural, outside of the agent, but gets progressively internalized within the agent via a combination of linguistic exposure and decision-making.

Exposed to a diversity of instructions, agents gain new cognitive abilities. The first is *abstraction*. Linguistic autotelic agents can reach and make sense of abstract relational goals "*sort objects by size*" (Jiang et al., 2019b), "*put the cylinder in the drawer*" (Lynch & Sermanet, 2021), sequential goals "*open the yellow door after opening a red door*" (Chevalier-Boisvert et al., 2019b), or even learning goals "*is the ghargh*

edible?" (Yuan et al., 2019). Whereas handling abstract goals used to require engineers to hard-code specific goal representations and reward functions within the agent (Colas et al., 2019a; Stooke et al., 2021), linguistic goals offer abstraction via simple linguistic interactions (Bahdanau et al., 2019b). Once abstractions have been distilled within the representations of the agent, they can be leveraged to augment its exploratory capacities. Searching for novelty in a space of abstract linguistic descriptions of the world is indeed more efficient than searching for novelty in low-level sensorimotor spaces which could be trivially triggered by leaves moving in the wind or TV noise (Tam et al., 2022; Mu et al., 2022).

A second cognitive ability is *systematic generalization*. Language-instructed agents indeed seem to demonstrate the ability to generalize to new instructions obtained by systematic recombinations of instructions they were trained on (Hill et al., 2020a). For instance, agents that learned to *grasp blue objects* and *put green objects on the table* can directly *grasp green objects* and *put blue objects on the table* (Hermann et al., 2017a; Chevalier-Boisvert et al., 2019b; Hill et al., 2020a,b; Sharma et al., 2021). This ability can either be encoded in learning architecture through the use of modular networks (neuro-symbolic approaches), or emerge spontaneously in plain networks under the right environmental conditions (Hill et al., 2020a). As we will show in chapter 8, sometimes the world does not conform to strict linguistic compositionality, systematic generalization still supports good priors — e.g. *feeding the cat* is not a strict transposition of *feeding the plant* but they still share similarities (bringing supplies to the cat/plant).

Learning to represent possible futures

After being exposed to aligned trajectories and linguistic descriptions, agents can generate concrete examples of abstract descriptions. The DECSTR approach, for example, trains a generative world model to sample from the distribution of possible future states matching a given abstract linguistic description (Akakzia et al., 2021b). This simple mapping supports *behavioral diversity*, the ability to represent different possible futures so as to select one to pursue. Similar setups could leverage DALL-E, an impressive text-to-image generative system (Ramesh et al., 2021, 2022). Trained on pairs of images and compositional descriptions, DALL-E can generate high-quality images from the most twisted descriptions humans can think of. The exposition to compositional language, paired with sufficiently powerful learning architectures and algorithms leads to impressive visual composition abilities that could be put to use to generate visual goals or to represent possible futures in embodied and situated agents.

Learning to decompose tasks

Vygotsky and others discovered that children's use of private speech helps them increase self-control and is instrumental to their capacity to reason and solve hard tasks (Vygotsky, 1934; Berk, 1994). The ability to formulate sentences like "at the left of the blue wall," for instance, predicts spatial orientation capacities in such contexts, while interfering with adult's inner speech via speaking tasks hinders theirs (Hermer-Vazquez, 2001).

Language indeed contains cues about how to decompose tasks into sub-tasks, i.e. how to *generate good plans*. Although *gharble* is a made-up word, *fry the gharble* probably involves preparation of the gharble (e.g. peeling, cutting), some sort of oil and a frying pan (Yuan et al., 2019). *Draw an octogon* contains cues about the decomposition of the task: *octo* means 8, so we should probably do something 8 times, etc. (Wong et al., 2021). Recent AI approaches leverage these regularities by training *plan generators* from linguistic task descriptions (Jiang et al., 2019b; Chen et al., 2021; Sharma et al., 2021; Mirchandani et al., 2021; Shridhar et al., 2021; Wong et al., 2021). Among them, Wong et al. use plan generation as an auxiliary task to train a drawing policy (Wong et al., 2021). Generating plans to solve a particular drawing task helps shape the internal representation of the main policy which, they find, favors abstraction and generalization in the main task. Interestingly, language only shapes representations and is not required at test time, in line with the requirement thesis of Dennett.

Inspired by video games of the 80s such as *Zork*, text-based environments define purely linguistic goals, actions, and states (Côté et al., 2018; Das et al., 2018; Yuan et al., 2019). Training a policy in such environments can be seen as training a plan generator in a linguistic world model, i.e. training an inner speech to generate good task decompositions. This idea was exploited in *AlfWorld*, where a pre-trained plan generator is deployed in a physical environment to generate sub-goals for a low-level policy (Shridhar et al., 2021). Here, the abstraction capabilities of language help the plan generator solve long-horizon tasks.

The above approaches echo the thesis of Dennett (Sec. 6.2): the mere exposure to structured language, once internalized within internal modules (reward function, policy, world model) strongly shapes inner representations in new ways and supports new cognitive functions (abstraction, future states generation, compositional generalization, task decomposition, etc).

Learning from cultural artifacts

Large language models (LLM) are trained on huge quantities of text scrapped from the internet: Wikipedia, forums, blogs, scientific articles, books, subtitles, etc. (Devlin et al., 2019; Brown et al., 2020b). As such, they can be seen as *cultural models* that contain information about our values, norms, customs, history, or interests (Hershcovich et al., 2022; Arora et al., 2022). This represents a great opportunity for autotelic agents to learn about us, align with us, and better navigate our complex world. So far, only very little research has leveraged that opportunity. An example is the use of a trained LLM to act as a zero-shot planner, i.e. a plan generator (Huang et al., 2022)/ Plugged with an interactive agent, the language model is used to generate sub-goals for the agent to solve the main task. Another work extracts information about complex time-extended behaviors from an LLM by asking it to score the actions available to the agent (Ahn et al., 2022). Finally, the MineDojo framework (Fan et al., 2022) proposes to caption thousands of YouTube videos of humans playing Minecraft using GPT-3 (Brown et al., 2020b), generating creative high-level tasks as well as low-level linguistic guidance for embodied agents.

6.4.2 Internalization of Language Production

Agents that internalize extractive models learn to exploit the information contained within linguistic vocabularies, structures and narratives. However, most of them require external linguistic inputs at test time and, thus, cannot be considered autonomous. Vygotskian autotelic agents reach autonomy by internalizing *productive models*; i.e. by learning to generate their own linguistic inputs, their own *inner speech* (see Fig. 6.1, c).

Inner speech can be understood as a fully-formed language: descriptions, explanations or advice to be fed back to extractive models; to serve as a common currency between cognitive modules (fully-formed inner speech) (Zeng et al., 2022). But it might also be understood as *distributed representations* within productive models, *upstream from fully-formed language* (distributed inner speech). In the latter interpretation, linguistic production acts as an auxiliary task whose true purpose is to shape the agent's cognitive representations. Symbolic behaviors might indeed not require explicit symbolic representations but may emerge from distributed architectures trained on structured tasks, e.g. involving linguistic predictions (McClelland et al., 2010; Santoro et al., 2021). In the literature, we found four types of productive models making use of either fully-formed or distributed inner speech: trajectory captioners, plan generators, explanation generators, and goal generators.

Trajectory captioners

Trajectory captioners are trained on instructive or descriptive feedback to generate valid descriptions of scenes or trajectories (Cideron et al., 2020b; Zhou & Small, 2020b; Nguyen et al., 2021; Carta et al., 2022; Yan et al., 2022). In line with Vygotsky's theory, these agents internalize models of descriptive social partners. They generate an *inner speech* describing their ongoing behaviors just like a caretaker would. Used as an auxiliary task (distributed inner speech), the generation of descriptions helps the agent shape its representation so as to generalize better to new tasks (Yan et al., 2022). With fully-formed inner speech, agents can generate new multi-modal data autonomously, and learn from past experience via *hindsight learning* (Andrychowicz et al., 2017b), i.e. the reinterpretation of their trajectory as a valid behavior to achieve the trajectory's description (Zhou & Small, 2020b; Nguyen et al., 2021).

Plan generators

Plan generators are both extractive and productive. Following the formalism of hierarchical RL (HRL), plan generators are implemented by a *high-level policy* generating linguistic sub-goals to a low-level policy (executioner) (Dayan & Hinton, 1993b; Sutton et al., 1999). Linguistic sub-goals are a form of inner speech that facilitates decision-making at lower temporal resolution by providing abstract, human-interpretable actions, which themselves favor systematic generalization for the low-level policy (see Sec. 6.4.1) (Jiang et al., 2019b; Chen et al., 2021; Shridhar et al., 2021). Here, agents internalize linguistic production to autonomously generate further guidance for themselves in fully-formed language (task decompositions).

Explanation generators

Vygotskian agents can generate *explanations*. Using the generation of explanations as an auxiliary task (distributed inner speech) was indeed shown to support causal and relational learning in complex *odd one out* tasks (Lampinen et al., 2022). Note however that this approach is neither embodied, nor autotelic.

Goal generators

Some forms of creativity appear easier in linguistic spaces because swapping words, compositing new sentences, and generating metaphors are all easier in the language space than in sensorimotor spaces. The IMAGINE approach, that we will detail in chapter 8, leverages this idea to support *creative goal imagination*. While previous methods were limited to generating goals within the distribution of past experience (e.g. with generative models of states (Nair et al., 2018b)), IMAGINE invents out-of-distribution goals by combining descriptions of past goals. These manipulations occur in linguistic spaces directly and are thus *linguistic thoughts*; fully-formed inner speech (Carruthers' view). The problem of goal imagination, difficult to solve in sensorimotor space, is projected onto the linguistic space, solved there, and projected back to sensorimotor space (Clark's view). This, in turn, powers additional cognitive abilities. First, it powers a creative exploration oriented towards objects and interactions with them. Second, it enhances systematic generalization by widening the set of goals the agent can train on.

By internalizing linguistic production, IMAGINE generates goals that are both *novel* (new sentences) and *appropriate* (they respect linguistic regularities, both structures, and contents) (Runco & Jaeger, 2012). Social descriptions focus on objects, object attributes, and interactions with these objects. Imagined goals obtained by recompositions of social ones share the same attentional and conceptual biases, e.g. by reusing semantic categories of a particular culture. Thus, cultural biases are implicitly transmitted to the agent, which forms goal representations and biases goal selection following cultural constraints.

Note that productive models are very rare in the literature. In the future, Vygotskian autotelic agents must learn to internalize productive models for all types of multi-modal feedback they encounter: advice, explanations, attention guidance, motivation, instructions, descriptions, etc. It is only by learning to generate this guidance for themselves that they may gain full control of their own behavior. The question of whether to use fully-formed or distributed inner speech remains open, as both strategies seem to find different use-cases. Because cultural models are biased, future agents will need to edit, correct, augment and generate their own interpretations of culture based on their individual experiences. How to efficiently steer language models in these ways remains a question to explore in future research.

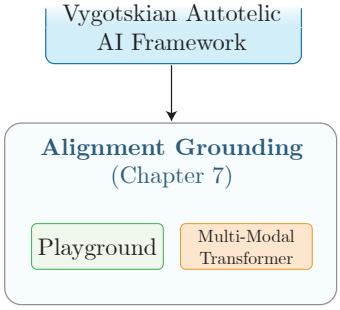
6.5 Conclusion

This chapter proposes a novel framework, the Vygotskian Autotelic AI, which builds upon the autotelic RL framework introduced in chapter 3 and draws inspiration from

Vygotsky's work. The framework advocates immersing artificial agents in complex socio-cultural environments to enable them to use their linguistic interactions as cognitive tools and enhance their learning abilities. To achieve optimal utilization of cultural conventions, agents must be equipped with two types of models: extractive and productive. Extractive models are conditioned by language and enable agents to refine their internal representations, leading to better abstraction and systematic generalization. Productive models, on the other hand, allow agents to generate feedback received from others, thereby creating new cognitive tools based on socio-cultural language. This chapter is followed by two computational studies. The first one, presented in chapter 7, examines the impact of inductive biases on extractive models of a Vygotskian autotelic agent and how they can facilitate systematic generalization. The second one, presented in chapter 8, introduces the IMAGINE agent, which leverages productive models to explore its environment creatively.

Chapter 7

Alignment: Grounding Spatio-Temporal Language with Transformers



Contents

In this chapter, we investigate the role of inductive biases in extractive models of Vygotksian autotelic agents (Fig. 6.1(b)). More specifically, we consider an autonomous embodied agent receiving linguistic descriptions of its behavioral traces and train a truth function that predicts if a description matches a given history of observations. The descriptions involve time-extended predicates in past and present tense as well as spatio-temporal references to objects in the scene. To study the role of architectural biases in this task, we train several models including multimodal Transformer architectures; the latter implement different attention computations between words and objects across space and time. We test models on two classes of generalization: 1) generalization to randomly held-out sentences; 2) generalization to grammar primitives.

7.1 Motivations

Embodied Language Grounding (Zwaan & Madden, 2005b) is the field that studies how agents can align language with their behaviors in order to extract the meaning of linguistic constructions. Early approaches in developmental robotics studied how various machine learning techniques, ranging from neural networks (Sugita & Tani, 2005; Tuci et al., 2011; Hinaut et al., 2014) to non-negative matrix factorization (Mangin et al., 2015), could enable the acquisition of grounded compositional language (Taniguchi et al., 2016; Tani, 2016). This line of work was recently extended using techniques for *Language conditioned Deep Reinforcement Learning* (Luketina et al., 2019). Among these works, we can distinguish mainly three language grounding strategies. The first one consists of directly grounding language in the behavior of agents by training goal-conditioned policies satisfying linguistic instructions (Sugita & Tani, 2005; Tuci et al., 2011; Hill et al., 2020a; Hermann et al., 2017b; Chaplot et al., 2018a). The second aims at extracting the meaning of sentences from mental simulations (i.e. generative models) of possible sensorimotor configurations matching linguistic descriptions (Mangin et al., 2015; Akakzia et al., 2021c; Cideron et al., 2020a; Nguyen et al., 2021). The third strategy searches to learn the meaning of linguistic constructs in terms of outcomes that agents can observe in the environment. This is achieved by training a truth function that detects if descriptions provided by an expert match certain world configurations. This truth function can be obtained via *Inverse Reinforcement Learning* (Zhou & Small, 2020a; Bahdanau et al., 2019b) or by training a multi-modal binary classifier.

While all the above-mentioned approaches consider language that describes immediate and instantaneous actions, we argue that it is also important for agents to grasp linguistic concepts that span multiple time scales. We thus propose to study the grounding of new spatio-temporal concepts enabling agents to ground time-extended predicates (Fig. 7.1a) with complex spatio-temporal references to objects (Fig. 7.1b) and understand both present and past tenses (Fig. 7.1c). To do so we choose the third strategy mentioned above, i.e. to train a truth function that predicts when descriptions match traces of experience. This choice is motivated by two important considerations. First, prior work showed that learning truth functions was key to fostering generalization (Bahdanau et al., 2019b), enabling agents to efficiently self-train policies goal relabeling (Cideron et al., 2020a) for instance. Hence the truth function is an important and self-contained compo-

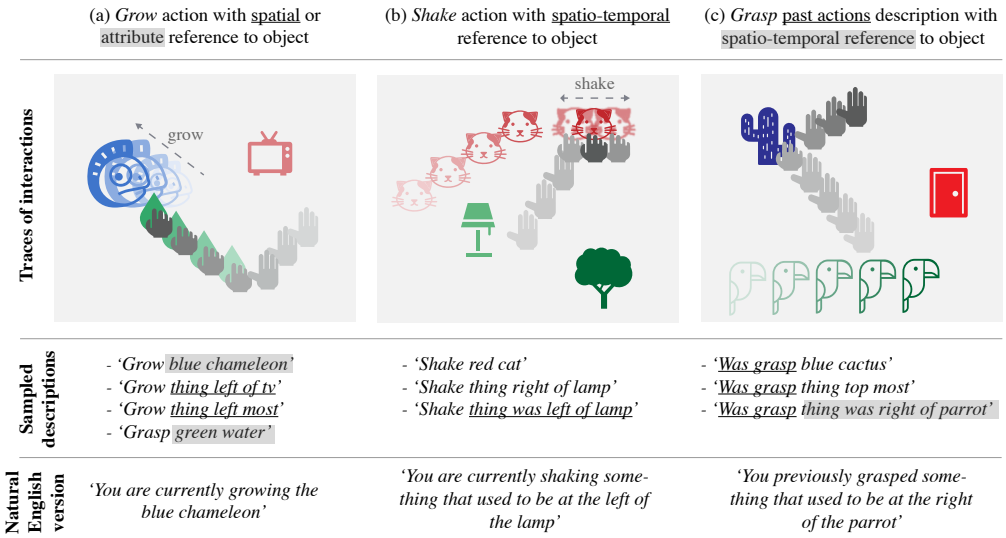


Figure 7.1: **Visual summary of the Temporal Playground environment:** At each episode (column a, b and c), the actions of an agent (represented by a hand) unfold in the environment and generate a trace of interactions between objects and the agent body. Given such a trace, the environment automatically generates a set of synthetic linguistic descriptions that are true at the end of the trace. In (a) the agent grows an object which is described with spatial (underlined) or attribute (highlighted) reference. In (b) it shakes an object which is described with attribute, spatial or spatio-temporal (underlined) reference. In (c) it has grasped an object (past action underlined) which is described with attribute, spatial or spatio-temporal (highlighted) reference.

nent of larger learning systems. Second, this strategy carefully controls the distribution of experiences and descriptions perceived by the agent.

Grounding spatio-temporal language is a relational problem. In the context of this paper, the concepts we aim at grounding are temporal and spatial, and thus relational by nature. But more generally, it is worth mentioning that embodied language grounding has a relational structure. We understand the meaning of words by analyzing the relations they state in the world (Gentner & Loewenstein, 2002). Actions are relations between subjects and objects and can be defined in terms of agent affordances (Gibson, 1968). As a result, we implement our truth function using relational architectures based on *Transformers* (Vaswani et al., 2017) and investigate the role of the relational bias (Battaglia et al., 2018) on learning. We propose a formalism unifying three variants of a multi-modal transformer inspired by Ding et al. (2020) that implement different relational operations. We measure the generalization capabilities of these architectures along three axis 1) generalization to new traces of experience; 2) generalization to randomly held out sentences; 3) generalization to grammar primitives, systematically held out from the training set as in Ruis et al. (2020). We observe that maintaining object identity in the attention computation of our Transformers is instrumental to achieving good performance on generalization overall. We also identify specific relational operations that are key to generalizing on certain grammar primitives.

Specific Contributions

In this chapter, we introduce:

1. *Playground*: a procedurally-generated environment designed to study several types of generalizations (across predicates, attributes, object types, and categories).
2. A new Embodied Language Grounding task focusing on spatio-temporal language;
3. A formalism unifying different relational architectures based on Transformers expressed as a function of mapping and aggregation operations;
4. A systematic study of the generalization capabilities of these architectures and the identification of key components for their success on this task.

7.2 The Playground Environment

The present study relies on behavioral trajectories of embodied artificial agents coupled with linguistic descriptions provided by a social partner. In this section, we first detail the physical interactions available to the agent. We then provide the grammar that is used by the programmatic social partner to describe the behavior of the agent[1].

7.2.1 The Environment

The environment is a 2D square: $[-1.2, 1.2]^2$. The agent is a disc of diameter 0.05 with an initial position $(0, 0)$. Objects have sizes uniformly sampled from $[0.2, 0.3]$ and their initial positions are randomized so that they are not in contact with each other. The agent has an action space of size 3 bounded in $[-1, 1]$. The first two actions control the agent's continuous 2D translation (bounded to 0.15 in any direction). The agent can grasp objects by getting in contact with them and closing its gripper (positive third action), unless it already has an object in hand. Objects include 10 animals, 10 plants, 10 pieces of furniture and 2 supplies. Admissible categories are *animal, plant, furniture, supply* and *living_thing* (animal or plant), see Fig. 7.2. Objects are assigned a color

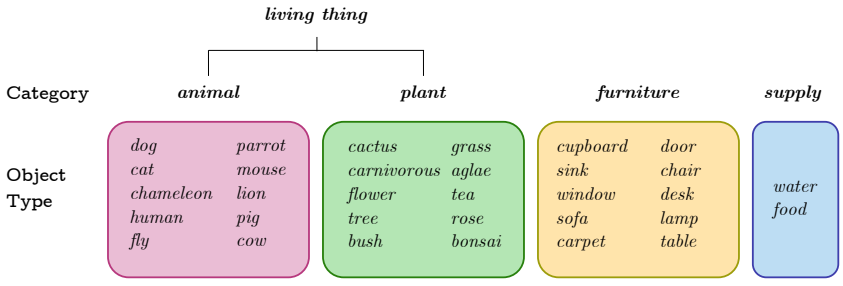


Figure 7.2: Representation of possible object types and categories.

attribute (red, blue, or green). Their precise color is a continuous RGB code uniformly

[1]Note that a simplified version of the playground environment will be used in chapter 8.

sampled from RGB subspaces associated with their attribute color. Each scene contains 3 of these procedurally-generated objects. When a supply is on an animal or water is on a plant (contact define as the distance between objects is equal to the mean size of the two objects $d = (size(obj_1) + size(obj_2))/2$), the object will grow over time with a constant growth rate until it reaches the maximum size allowed for objects or until contact is lost.

7.2.2 The Temporal Grammar

To enable a controlled and systematic study of how different types of spatio-temporal linguistic meanings can be learned, we argue it is necessary to first conduct a systematic study with a controlled synthetic grammar. We thus consider a synthetic language with a vocabulary of size 53 and sentences with a maximum length of 8. This synthetic language facilitates the generation of descriptions matching the behavioral traces of the agent. The synthetic language we use can be decomposed into two components: the instantaneous grammar and the temporal logic below:

Instantaenous grammar.

```
            <S>::=   <pred> <thing_A>
         <pred>::=   grow | grasp | shake
      <thing_A>::=   <thing_B> | <attr> <thing_B> | thing <localizer> | thing <localizer_all>
    <localizer>::=   left of <thing_B> | right of <thing_B> | top of <thing_B> | bottom of <thing_B>
<localizer_all>::=   left most| right most| top most| bottom most
      <thing_B>::=   dog | cactus | ...| living_thing | thing
         <attr>::=   blue | green | red
```

Note that although the sentence "*Grow red door*" is valid in the grammar, it will never be communicated by the social partner as pieces of furniture cannot grow.

Temporal logic.

```
      <S>::=   was <pred> <thing_A>
<thing_A>::=   thing was <localizer> | thing was <localizer_all>
```

7.2.3 Concept Definition

We split the set of all possible descriptions output by our grammar into four conceptual categories according to the rules given in Table 7.1. The four concepts are:

- **Sentences involving basic concepts.** This category of sentences talk about present-time events by referring to objects and their attributes. Sentences begin with the *'grasp'* token combined with any object. Objects can be named after their category (eg. *'animal', 'thing'*) or directly by their type (*'dog', 'door', 'algae', etc.*). Finally, the color *('red','blue','green'*) of objects can also be specified.
- **Sentences involving spatial concepts.** This category of sentences additionally involve one-to-one spatial relations and one-to-all spatial relations to refer to objects. An object can be *'left of'* another object (reference is made in relation to a

single other object), or can be the *'top most'* object (reference is made in relation with all other objects). Example sentences include *'grasp thing bottom of cat'* or *'grasp thing right most'*.

- **Sentences involving temporal concepts**. This category of sentences involves talking about temporally-extended predicates and the past tense, without any spatial relations. The two temporal predicates are denoted with the words *'grow'* and *'shake'*. The truth value of these predicates can only be decided by looking at the temporal evolution of the object's size and position respectively. A predicate is transposed at the past tense if the action it describes was true at some point in the past and is no longer true in the present, this is indicated by adding the modifier *'was'* before the predicate. Example sentences include *'was grasp red chameleon'* (indicating that the agent grasped the red chameleon and then released it) and *'shake bush'*;
- **Sentences involving spatio-temporal concepts**. Finally, we consider the broad class of spatio-temporal sentences that combine spatial reference and temporal or past-tense predicates. These are sentences that involve both the spatial and temporal concepts defined above. Additionally, there is a case of where the spatial and the temporal aspects are entangled: past spatial reference. This happens when an object is referred to by its previous spatial relationship with another object. Consider the case of an animal that was at first on the bottom of a table, then moved on top, and then is grasped. In this case we could refer to this animal as something that was previously on the bottom of the table. We use the same *'was'* modifier as for the past tense predicates; and thus we would describe the action as *'Grasp thing was bottom of table'*.

Concept	BNF	Size
1. Basic	<S> ::= <pred> <thing_A> <pred> ::= *grasp* <thing_A> ::= <thing_B> | <attr> <thing_B>	152
2. Spatial	<S> ::= <pred> <thing_A> <pred> ::= *grasp* <thing_A> ::= <*thing* <localizer> | *thing* <localizer_all>	156
3. Temporal	<S> ::= <pred_A> <thing_A> | *was* <pred_B> <thing_A> <pred_A> ::= *grow* | *shake* <pred_B> ::= *grasp* | *grow* | *shake* <thing_A> ::= <thing_B> | <attr> <thing_B>	648
4. Spatio-Temporal	<S> ::= <pred_A> <thing_A> | *was* <pred_B> <thing_A> <pred_C> <thing_C> <pred_A> ::= *grow* | *shake* <pred_B> ::= *grasp* | *grow* | *shake* <pred_C> ::= *grasp* <thing_A> ::= *thing* <localizer> | *thing* <localizer_all> | *thing was* <localizer> | *thing was* <localizer_all> | <thing_C> ::= *thing was* <localizer> | *thing was* <localizer_all>	1716

Table 7.1: **Concept categories with their associated BNF.** <thing_B>, <attr>, <localizer> and <localizer_all>.

7.2.4 Data generation

Traces Generation. To generate traces matching the descriptions we use a scripted bot. The bot uses a hand-defined predicate-conditioned policy (*grasp*, *grow* and *shake*) and performs rollouts in the environment. The policies are then conditioned on a boolean variable that modulates the behavior to obtain a mix of predicates in the present and the past tenses. For instance, if a *grasp* policy is used, there will be a 50% chance that the scenario will end with the object being grasped, leading to a present-tense description; and a 50% chance that the agent releases the object, yielding a past tense description.

Description generation. For each time step, the instantaneous grammar generates the set of all true instantaneous sentences using a set of filtering operations similar to the one used in CLEVR (Johnson et al., 2016), without the past predicates and past spatial relations. Then the temporal logic component uses these linguistic traces in the following way: if a given sentence for a predicate is true in a past time step and false in the present time step, the prefix token *'was'* is prepended to the sentence; similarly, if a given spatial relation is observed in a previous time step and unobserved in the present, the prefix token *'was'* is prepended to the spatial relation.

The data collected consists of 56837 trajectories of $T = 30$ time steps. Among the traces some descriptions are less frequent than others but we make sure to have at least 50 traces representing each of the 2672 descriptions we consider. We record the observed episodes and sentences in a buffer, and when training a model we sample (S, W, r) tuples with one observation coupled with either a true sentence from the buffer or another false sentence generated from the grammar.

7.3 Problem

We consider the setting of an embodied agent behaving in an environment. This agent interacts with the surrounding objects over time, during an episode of fixed length (T). Once this episode is over, an oracle provides exhaustive feedback in a synthetic language about everything that has happened. This language describes actions of the agent over the objects and includes spatial and temporal concepts. The spatial concepts are a reference to an object through its spatial relation with others (Fig. 7.1a), and the temporal concepts are the past modality for the actions of the agent (Fig. 7.1c), past modality for spatial relations (Fig. 7.1b), and actions that unfold over time intervals. The histories of states of the agent's body and of the objects over the episode as well as the associated sentences are recorded in a buffer $\mathcal{B}$. From this setting, and echoing previous work on training agents from descriptions, we frame the Embodied Language Grounding problem as learning a parametrized truth function R_θ over couples of observations traces and sentences, tasked with predicting whether a given sentence W is true of a given episode history S or not. Formally, we aim to minimize

$$\mathbb{E}_{(S,W)\sim\mathcal{B}}\big[\mathcal{L}(R_\theta(S,W), r(S,W))\big]$$

where $\mathcal{L}$ denotes the cross-entropy loss and r denotes the ground truth boolean value for sentence W about trace S.

7.4 Multi-modal Transformers

7.4.1 Neural Network Architectures

In this section we describe the architectures used as well as their inputs. Let one input sample to our model be $I = (S, W)$, where $(S_{i,t})_{i,t}$ represents the objects' and body's evolution, and $(W_l)_l$ represents the linguistic observations. S has a spatial (or entity) dimension indexed by $i \in [0..N]$ and a temporal dimension indexed by $t \in [1..T]$; for any i, t, $S_{i,t}$ is a vector of observational features. Note that by convention, the trace $(S_{0,t})_t$ represents the body's features, and the traces $(S_{i,t})_{t,i>0}$ represents the other objects' features. W is a 2-dimensional tensor indexed by the sequence $l \in [1..L]$; for any l, $W_l \in \mathbb{R}^{dw}$ is a one-hot vector defining the word in the dictionary. The output to our models is a single scalar between 0 and 1 representing the probability that the sentence encoded by W is true in the observation trace S.

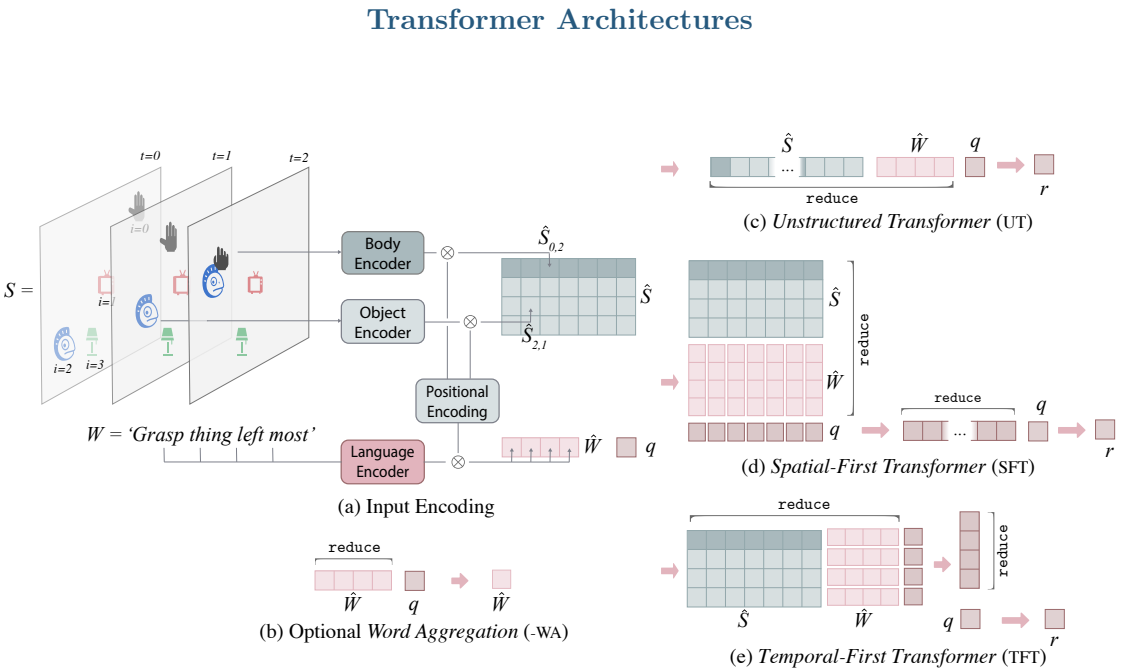


Figure 7.3: **Visual summary of the architectures used.** We show the details of UT, SFT and TFT respectively in subfigures (c), (d), (e), as well as a schematic illustration of the preprocessing phase (a) and the optional word-aggregation procedure (b).

To systematically study the influence of architectural choices on language performance and generalization in our spatio-temporal grounded language context, we define a set of mapping and aggregation operations that allows us to succinctly describe different models in a unified framework. We define:

- An aggregation operation based on a Transformer model, called `reduce`. `reduce` is a parametrized function that takes 3 inputs: a tensor, a dimension tuple D over which to reduce and a query tensor (that has to have the size of the reduced

tensor). R layers of a Transformer are applied to the input-query concatenation and are then queried at the position corresponding to the query tokens. This produces an output reduced over the dimensions D.

- A casting operation called `cast`. `cast` takes as input 2 tensors A and B and a dimension d. A is flattened, expanded so as to fit the tensor B in all dimensions except d, and concatenated along the d dimension.
- A helper expand operation called `expand` that takes as arguments a tensor and an integer n and repeats the tensor n times.

Using those operations, we define three architectures: one with no particular bias (*Unstructured Transformer*, inspired by Ding et al. (2020), or UT); one with a spatial-first structural bias – objects and words are aggregated along the spatial dimension first (*Spatial-First Transformer* or SFT); and one with a temporal-first structural bias – objects and words are aggregated along the temporal dimension first (*Temporal-First Transformer*, or TFT).

Before inputting the observations of bodies and objects S and the language W into any of the Transformer architectures, they are projected to a common dimension (see Sup. Section C.1.1 for more details). A positional encoding (Vaswani et al., 2017) is then added along the time dimension for observations and along the sequence dimension for language; and finally a one-hot vector indicating whether the vector is observational or linguistic is appended at the end. This produces the modified observation-language tuple $(\hat{S}, \hat{W})$. We let:

$$\text{UT}(\hat{S}, \hat{W}) := \texttt{reduce}(\texttt{cast}(\hat{S}, \hat{W}, 0), 0, q)$$

$$\text{SFT}(\hat{S}, \hat{W}, q) := \texttt{reduce}(\texttt{reduce}(\texttt{cast}(\hat{W}, \hat{S}, 0), 0, \texttt{expand}(q, T)), 0, q)$$

$$\text{TFT}(\hat{S}, \hat{W}, q) := \texttt{reduce}(\texttt{reduce}(\texttt{cast}(\hat{W}, \hat{S}, 1), 1, \texttt{expand}(q, N+1)), 0, q)$$

where T is the number of time steps, N is the number of objects and q is a learned query token. See Fig. 7.3 for an illustration of these architectures.

Note that SFT and TFT are transpose versions of each other: SFT is performing aggregation over space first and then time, and the reverse is true for TFT. Additionally, we define a variant of each of these architectures where the words are aggregated before being related with the observations. We name these variants by appendding -WA (word-aggregation) to the name of the model (see Fig. 7.3 (b)).

$$\hat{W} \leftarrow \texttt{reduce}(\hat{W}, 0, q)$$

We examine these variants to study the effect of letting word-tokens directly interact with object-token through the self-attention layers vs simply aggregating all language tokens in a single embedding and letting this vector condition the processing of observations. The latter is commonly done in the language-conditioned RL and language grounding literature (Chevalier-Boisvert et al., 2019c; Bahdanau et al., 2019b; Hui et al., 2020; Ruis et al., 2020), using the language embedding in FiLM layers (Perez et al., 2017) for instance. Finding a significant effect here would encourage using architectures which allow direct interactions between the word tokens and the objects they refer to.

LSTM Baselines

We also compare some LSTM-based baselines on this task:

1. LSTM-FLAT: This variant has two internal LSTM: one that processes the language and one that processes the scenes as concatenations of all the body and object features. This produces two vectors that are concatenated into one, which is then run through an MLP and a final softmax to produce the final output.
2. LSTM-FACTORED: This variant independently processes the different body and object traces, which have previously been projected to the same dimension using a separate linear projection for the object and for the body. The language is processed by a separate LSTM. These body, object and language vectors are finally concatenated and fed to a final MLP and a softmax to produce the output.

7.4.2 Training and Testing Procedures

For each of the Transformer variants (6 models) and the LSTM baselines (2 models) we perform an hyper parameter search using 3 seeds in order to extract the best configuration. We extract the best condition for each model by measuring the mean F_1 on a testing set made of uniformly sampled descriptions from each of the categories define in Sec. 7.2. We use the F_1 score because testing sets are imbalanced (the number of traces fulfilling each description is low). We then retrain best configurations over 10 seeds and report the mean and standard deviation (reported as solid black lines in Fig. 7.4 and Fig. 7.5) of the averaged F_1 score computed on each set of sentences. When statistical significance is reported in the text, it is systematically computed using a two-tail Welch's t-test with null hypothesis $\mu_1 = \mu_2$, at level $\alpha = 0.05$ (Colas et al., 2019b). Details about the training procedure and the hyper parameter search are provided in Sup. Section C.1.2.

7.5 Experiments

7.5.1 Generalization Abilities of Models on Non-Systematic Split by Categories of Meaning

In this experiment, we perform a study of generalization to new sentences from known observations. We divide our set of test sentences in four categories based on the categories of meanings listed in Sec. 7.2: Basic, Spatial, Spatio-Temporal and Temporal. We remove 15% of all possible sentences in each category from the train set and evaluate the F1 score on those sentences. The results are provided in Fig. 7.4.

First, we notice that over all categories of meanings, all UT and TFT models, with or without word-aggregation, perform extremely well compared to the LSTM baselines, with all these four models achieving near-perfect test performance on the Basic sentences, with very little variability across the 10 seeds. We then notice that all SFT variants perform poorly on all test categories, in line or worse than the baselines. This is particularly visible on the spatio-temporal category, where the SFT models perform at 0.75 ± 0.020 whereas the baselines perform at 0.80 ± 0.019 . This suggests that across tasks, it is

harmful to aggregate each scene plus the language information into a single vector. This may be due to the fact that objects lose their identity in this process, since information about all the objects becomes encoded in the same vector. This may make it difficult for the network to perform computations about the truth value of predicate on a single object.

Secondly, we notice that the word-aggregation condition seems to have little effect on the performance on all three Transformer models. We only observe a significant effect for UT models on spatio-temporal concepts (p-value = 2.38e-10). This suggests that the meaning of sentences can be adequately summarised by a single vector; while maintaining separated representations for each object is important for achieving good performance it seems unnecessary to do the same for linguistic input. However we notice during our hyperparameter search that our -WA models are not very robust to hyperparameter choice, with bigger variants more sensitive to the learning rate.

Thirdly, we observe that for our best-performing models, the basic categories of meanings are the easiest, with a mean score of 1.0 ± 0.003 across all UT and TFT models, then the spatial ones at 0.96 ± 0.020, then the temporal ones at 0.96 ± 0.009, and finally the spatio-temporal ones at 0.89 ± 0.027. This effectively suggests, as we hypothesised, that sentences containing spatial relations or temporal concepts are harder to ground than those who do not.

Known sentences with novel observations

We also examine the mean performance of our models for sentences in the training set but evaluated on a set of *new observations*: we generate a new set of rollouts on the environment, and only evaluate the model on sentences seen at train time (plots are reported in Sup. Section C.2). We see the performance is slightly better in this case, especially for the LSTM baselines (0.82 ± 0.031 versus 0.79 ± 0.032), but the results are comparable in both cases, suggesting that the main difficulty for models lies in grounding spatio-temporal meanings and not in linguistic generalization for the type of generalization considered in this section.

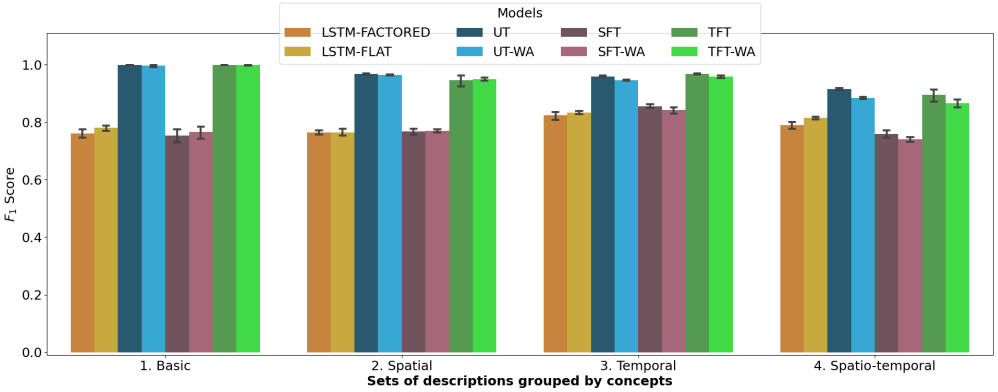


Figure 7.4: **F1 scores for all the models on randomly held-out sentences.** F_1 is measured on separated sets representing each category of concepts defined in Sec. 7.2.

7.5.2 Systematic Generalization on Withheld Combinations of Words

In addition to the previous generalization studies, we perform an experiment in a harder linguistic generalization setting where we systematically remove binary combinations in our train set. This is in line with previous work on systematic generalization on deep learning models (Lake & Baroni, 2018; Ruis et al., 2020; Hupkes et al., 2020). We create five test sets to examine the abilities of our models to generalize on binary combinations of words that have been systematically removed from the set of training sentences, but whose components have been seen before in other contexts. Our splits can be described by the set of forbidden combinations of words as:

1. **Forbidden object-attribute combinations.** remove from the train set all sentences containing *'red cat'*, *'blue door'* and *'green cactus'*. This tests the ability of models to recombine known objects with known attributes;
2. **Forbidden predicate-object combination.** remove all sentences containing *'grow'* and all objects from the *'plant'* category. This tests the model's ability to apply a known predicate to a known object in a new combination;
3. **Forbidden one-to-one relation.** remove all sentences containing *'right of'*. Since the *'right'* token is already seen as-is in the context of one-to-all relations (*'right most'*), and other one-to-one relations are observed during training, this tests the abilities of models to recombine known directions with in a known template;
4. **Forbidden past spatial relation.** remove all sentences containing the contiguous tokens *'was left of'*. This tests the abilities of models to transfer a known relation to the past modality, knowing other spatial relations in the past;
5. **Forbidden past predicate.** remove all sentences containing the contiguous tokens *'was grasp'*. This tests the ability of the model to transfer a known predicate to the past modality, knowing that it has already been trained on other past-tense predicates.

To avoid retraining all models for each split, we create one single train set with all forbidden sentences removed and we test separately on all splits. We use the same hyperparameters for all models than in the previous experiments. The results are reported in Fig. 7.5.

First we can notice that the good test scores obtained by the UT and TFT models on the previous sections are confirmed in on this experiment: they are the best performing models overall. We then notice that the first two splits, corresponding to new attribute-object and predicate-object combinations, are solved by the UT and TFT models, while the SFT models and the LSTM baselines struggle to achieve high scores. For the next 3 splits, which imply new spatial and temporal combinations, the scores overall drop significantly; we also observe much wider variability between seeds for each model, perhaps suggesting the various strategies adopted by the models to fit the train set have very different implications in terms of systematic generalization on spatial and temporal concepts. This very high variability between seeds on systematic generalization scores are reminiscent of the results obtained on the gSCAN benchmark (Ruis et al., 2020).

Additionally, for split 3, which implies combining known tokens to form a new spatial relation, we observe a significant drop in generalization for the word-aggregation (WA)

conditions, consistent across models (on average across seeds, -0.14 ± 0.093, -0.15 ± 0.234 and -0.20 ± 0.061 for UT, SFT and TFT resp. with p-values $<$ 1e-04 for UT and SFT). This may be due to the fact that recombining any one-to-one relation with the known token *right* seen in the context of one-to-all relations requires a separate representation for each of the linguistic tokens. The same significant drop in performance for the WA condition can be observed for UT and TFT in split 4, which implies transferring a known spatial relation to the past.

However, very surprisingly, for split 5 – which implies transposing the known predicate *grasp* to the past tense – we observe a very strong effect in the opposite direction: the WA condition seems to help generalizing to this unknown past predicate (from close-to-zero scores for all transformer models, the *WA* adds on average 0.71 ± 0.186, 0.45 ± 0.178 and 0.52 ± 0.183 points for UT, ST and TT resp. and p-values$<$ 1e-05). This may be due to the fact that models without WA learn a direct and systematic relationship between the *grasp* token and grasped objects, as indicated in their features; this relation is not modulated by the addition of the *was* modifier as a prefix to the sentence. Models do not exhibit the same behavior on split 4, which has similar structure (transfer the relation *left of* to the past). This may be due to the lack of variability in instantaneous predicates (only the *grasp* predicate); whereas there are several spatial relations (4 one-to-one, 4 one-to-all).

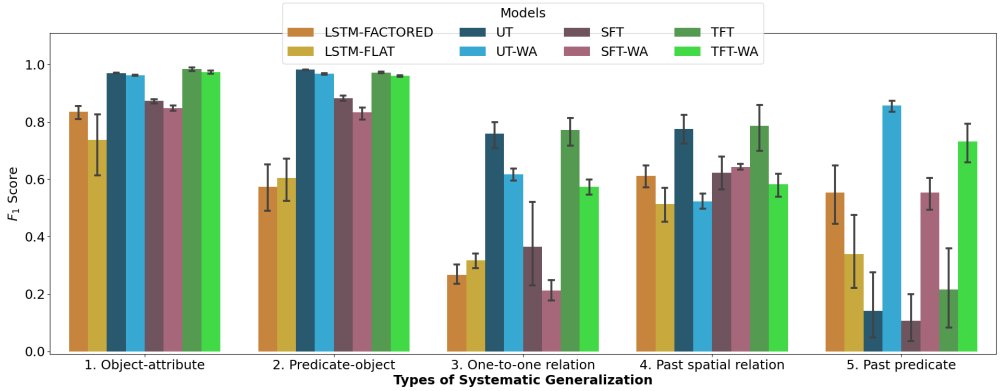


Figure 7.5: **F_1 scores of all the models on systematic generalization splits.** F_1 is measured on separated sets representing each of the forbidden combinations of word defined above.

7.6 Related Work

The idea that agents should learn to represent and ground language in their experience of the world has a long history in developmental robotics (Zwaan & Madden, 2005b; Steels, 2006; Sugita & Tani, 2005; Cangelosi et al., 2010b) and was recently extended in the context of Language Conditioned Deep Reinforcement Learning (Chevalier-Boisvert et al., 2019c; Hermann et al., 2017a; Luketina et al., 2019; Bahdanau et al., 2019b). These recent approaches often consider navigation (Chen & Mooney, 2011b; Chaplot et al., 2018b) or object manipulation (Akakzia et al., 2021c; Hermann et al., 2017a) tasks

and are always using instructive language. Meanings typically refer to instantaneous actions and rarely consider spatial reference to objects (Paul et al., 2016). Although our environment includes object manipulations, we here tackle novel categories of meanings involving the grounding of spatio-temporal concepts such as the past modality or complex spatio-temporal reference to objects.

We evaluate our learning architectures on their ability to generalise to sets of descriptions that contain systematic differences with the training data so as to assess whether they correctly model grammar primitives. This procedure is similar to the *gSCAN* benchmark (Ruis et al., 2020). This kind of compositional generalisation is referred as 'systematicity' by Hupkes et al. (2020). Environmental drivers that facilitate systematic generalization are also studied by Hill et al. (2020a). Although Hupkes et al. (2020) consider relational models in their work, they do not evaluate their performance on a *Language Grounding* task. Ruis et al. (2020) consider an Embodied Language Grounding setup involving one form of time-extended meanings (adverbs), but do not consider the past modality and spatio-temporal reference to objects, and do not consider learning truth functions. Also, they do not consider learning architectures that process sequences of sensorimotor observations. To our knowledge, no previous work has conducted systematic generalization studies on an Embodied Language Grounding task involving spatio-temporal language with Transformers.

The idea that relational architectures are relevant models for Language Grounding has been previously explored in the context of *Visual Reasoning*. They were indeed successfully applied for spatial reasoning in the visual question answering task *CLEVR* (Santoro et al., 2017). With the recent publication of the video reasoning dataset *CLEVRER* (Yi et al., 2020), those models were extended and demonstrated abilities to reason over spatio-temporal concepts, correctly answering causal, predictive and counterfactual questions (Ding et al., 2020). In contrast to our study, these works around CLEVRER do not aim to analyze spatio-temporal language and therefore do not consider time-extended predicates or spatio-temporal reference to objects in their language, and do not study properties of systematic generalization over sets of new sentences.

7.7 Discussion

In this work, we have presented a first step towards learning Embodied Language Grounding of spatio-temporal concepts, framed as the problem of learning a truth function that can predict if a given sentence is true of temporally-extended observations of an agent interacting with a collection of objects. We have studied the impact of architectural choices on successful grounding of our artificial spatio-temporal language. We have modelled different possible choices for aggregation of observations and language as hierarchical Transformer architectures. We have demonstrated that in our setting, it is beneficial to process temporally-extended observations and language tokens side-by-side, as evidenced by the good score of our Unstructured Transformer variant. However, there seems to be only minimal effect on performance in aggregating temporal observations along the temporal dimension first – compared to processing all traces and the language in an unstructured manner – as long as object identity is preserved. This can inform architectural design in cases where longer episode lengths make it impossible to store all individual timesteps

for each object; our experiments provide evidence that a temporal summary can be used in these cases. Our experiments with systematic dimensions of generalization provide mixed evidence for the influence of summarizing individual words into a single vector, showing it can be detrimental to generalize to novel word combinations but also can help prevent overgeneralization of a relation between a single word and a single object without considering the surrounding linguistic context.

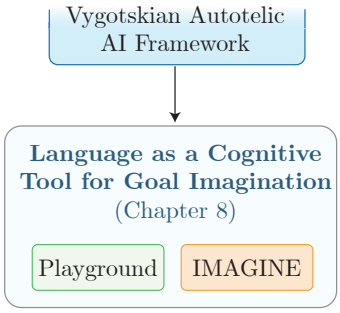
Limitations and further work.

There are several limitations of our setup which open important opportunities for further work. First, we have used a synthetic language that could be extended: for instance with more spatial relations and relations that are more than binary. Another axis for further research is using low-level observations. In our setting, we wanted to disentangle the effect of structural biases on learning spatio-temporal language from the problem of extracting objects from low level observations (Burgess et al., 2019; Greff et al., 2020; Engelcke et al., 2020; Locatello et al., 2020; Carion et al., 2020) in a consistent manner over time (object permanence (Creswell et al., 2020; Zhou et al., 2021). Further steps in this direction are needed, and it could allow us to define richer attributes (related to material or texture) and richer temporal predicates (such as breaking, floating, etc). Finally, we use a synthetic language which is far from the richness of the natural language used by humans, but previous work has shown that natural language can be projected onto the subspace defined by synthetic language using the semantic embeddings learned by large language models (Marzoev et al., 2020): this opens up be a fruitful avenue for further investigation.

In the next chapter, we will present the IMAGINE agent which grounds the meaning of descriptions provided by a social partner using a similar reward function as the one developed in this chapter. In addition to a reward function, the IMAGINE agent will be equipped with a language-conditioned policy enabling it to convert social descriptions into targetable goals and reach them. Finally, the IMAGINE agent has a productive model that leverages language compositionality to imagine new goals never communicated by the social partner.

Chapter 8

Language as a Cognitive Tool to Imagine Goals in Curiosity Driven Exploration: IMAGINE



Contents

This chapter presents IMAGINE, the first explicit Vygotksian autotelic agent that trains both extractive and productive models. First, IMAGINE leverages social interactions to convert linguistic descriptions of its behavior into targetable goals. Second, IMAGINE trains a reward function that assesses when it reaches a certain configuration expressed by a description (as in chapter 7). Then it utilizes the learned reward function to train a policy that reaches the linguistic goals. Once it has mastered a certain number of linguistic goals, it learns to recombine them in new ways in order to invent novel plans. This mechanisms powers a creative exploration. We decide to study it in a simplified version of the playground environment presented in chapter 7.

8.1 Motivations

The origin of this study stems from the observation that the majority of autotelic RL agents (presented in 3.3.1) have a limitation in that they can only learn to solve a limited range of concrete goals, that are not diverse. Indeed, the most standard approach is often to learn goal representation by training a variational autoencoder (VAE) of visual states (Nair et al., 2018b; Pong et al., 2020; Laversanne-Finot et al., 2018). This technic is very powerful because it offers both the possibility to learn to represent and sample goals. However, because goals are just a projection of states they lack abstraction. Crucially, the novel goals sampled from the latent space are concrete states that are within the distribution of already discovered effects. The challenge is therefore to achieve more abstract goal representation and to move beyond *within-distribution* goal generation.

In this difficult task, children leverage the properties of language to assimilate thousands of years of experience embedded in their culture, in only a few years (Tomasello, 1999a; Bruner, 1991). As detailed in chapter 6, language does not only offer humans the capacity to represent abstract concepts, it also enables them to manipulate and compose them to produce new plans. Interestingly, this generative capability can push the limits of the real, as illustrated by Chomsky (1957a)'s famous example of a sentence that is syntactically correct but semantically original "*Colorless green ideas sleep furiously*". Language can thus be used to generate out-of-distributions goals by leveraging compositionality to imagine new goals from known ones.

This chapter presents **I**ntrinsic **M**otivations **A**nd **G**oal **IN**vention for **E**xploration (IMAGINE): a learning architecture which leverages natural language (NL) interactions with a descriptive social partner (SP) to explore procedurally-generated scenes and interact with objects. IMAGINE discovers meaningful environment interactions through its own exploration (Fig. 8.1a) and episode-level NL descriptions provided by SP (8.1b). These descriptions are turned into targetable goals by the agent (8.1c). The agent learns to represent goals by jointly training a language encoder mapping NL to goal embeddings and a goal-achievement reward function (8.1d). The latter evaluates whether the current scene satisfies any given goal. These signals (ticks in Fig. 8.1d-e) are then used as training signals for policy learning. More importantly, IMAGINE can invent new goals by

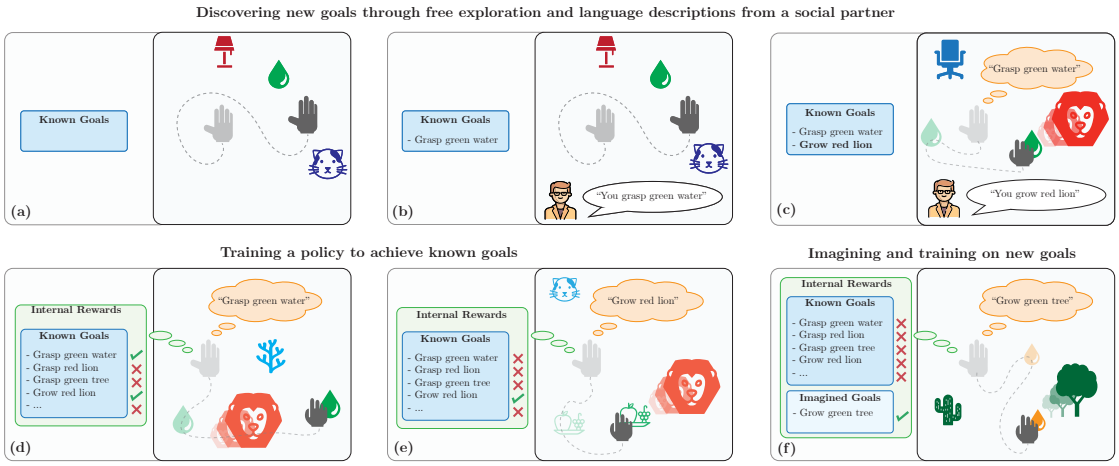


Figure 8.1: **IMAGINE overview**. In the *Playground* environment, the agent (hand) can move, grasp objects and grow some of them. Scenes are generated procedurally with objects of different types, colors and sizes. A social partner provides descriptive feedback (orange), that the agent converts into targetable goals (red bubbles).

composing known ones (8.1f). Its internal goal-achievement function allows it to train autonomously on these imagined goals.

Related work

The idea that language understanding is grounded in one's experience of the world and should not be secluded from the perceptual and motor systems has a long history in Cognitive Science (Glenberg & Kaschak, 2002a; Zwaan & Madden, 2005a). This vision was transposed to intelligent systems (Steels, 2006; McClelland et al., 2020), applied to human-machine interaction (Dominey, 2005; Madden et al., 2010) and recently to deep RL via frameworks such as *BabyAI* (Chevalier-Boisvert et al., 2019a).

In their review of *RL algorithms informed by NL*, Luketina et al. (2019) distinguish between *language-conditional* problems where language is required to solve the task and *language-assisted* problems where language is a supplementary help. In the first category, most works propose instruction-following agents (Branavan et al., 2010; Chen & Mooney, 2011a; Bahdanau et al., 2019a; Co-Reyes et al., 2019; Jiang et al., 2019a; Goyal et al., 2019; Cideron et al., 2020c). Although our system is *language-conditioned*, it is not *language-instructed*: it is never given any instruction or reward but sets its own goals and learns its own internal reward function. Bahdanau et al. (2019a) and Fu et al. (2019) also learn a reward function but require extensive expert knowledge (expert dataset and known environment dynamics respectively), whereas our agent uses experience generated by its own exploration.

Language is also particularly well suited for Hindsight Experience Replay (Andrychowicz et al., 2017a): descriptions of the current state can be used to relabel trajectories, enabling agents to transfer skills across goals. While previous works used a hard-coded descriptive function (Chan et al., 2019a; Jiang et al., 2019a) or trained a generative

model (Cideron et al., 2020c) to generate goal substitutes, we leverage the learned reward function to scan goal candidates.

To our knowledge, no previous work has considered the use of compositional goal imagination to enable creative exploration of the environment. The linguistic basis of our goal imagination mechanism is grounded in construction grammar (CG). CG is a usage-based approach that characterizes language acquisition as a trajectory starting with pattern imitation and the discovery of equivalence classes for argument substitution, before evolving towards the recognition and composition of more abstract patterns (Tomasello, 2000; Goldberg, 2003). This results in a structured inventory of constructions as form-to-meaning mappings that can be combined to create novel utterances (Goldberg, 2003). The discovery and substitution of equivalent words in learned schemas is observed directly in studies of child language (Tomasello & Olguin, 1993; Tomasello, 2000). Computational implementations of this approach have demonstrated its ability to foster generalization (Hinaut & Dominey, 2013) and was also used for data augmentation to improve the performance of neural seq2seq models in NLP (Andreas, 2020).

Imagining goals by composing known ones only works in association with *systematic generalization* (Bahdanau et al., 2019c; Hill et al., 2020a): generalizations of the type *grow any animal + grasp any plant → grow any plant*. These were found to emerge in instruction-following agents, including generalizations to new combinations of motor predicates, object colors and shapes (Hermann et al., 2017a; Hill et al., 2020a; Bahdanau et al., 2019a). Systematic generalization can occur when objects share common attributes (e.g. type, color). We directly encode that assumption into our models by representing objects as *single-slot object files* (Green & Quilty-Dunn, 2017): separate entities characterized by shared attributes. Because all objects have similar features, we introduce a new object-centered inductive bias: object-based modular architectures based on Deep Sets (Zaheer et al., 2017).

Specific Contributions

This chapter introduces:

1. The concept of imagining new goals using language compositionality to drive exploration.
2. IMAGINE: an intrinsically motivated agent that uses goal imagination to explore its environment, discover and master object interactions by leveraging NL descriptions from a social partner.
3. Modular policy and reward function with systematic generalization properties enabling IMAGINE to train on imagined goals. Modularity is based on Deep Sets, gated attention mechanisms and object-centered representations.
4. A study of IMAGINE investigating: 1) the effects of our goal imagination mechanism on generalization and exploration; 2) the identification of general properties of imagined goals required for any algorithm to have a similar impact; 3) the impact of modularity and 4) social interactions.

8.2 Problem Defintion

8.2.1 Open-ended Learning in a Socio-cultural Environment

We consider a setup where agents evolve in an environment filled with objects and have no prior on the set of possible interactions. An agent decides what and when to learn by setting its own goals and has no access to external rewards.

However, to allow the agent to learn relevant skills, a social partner (SP) can watch the scene and plays the role of a human caregiver. Following a developmental approach (Asada et al., 2009), we propose a hard-coded surrogate SP that models important aspects of the developmental processes seen in humans:

- At the beginning of each episode (left of Fig. 8.2), the agent chooses a goal by formulating a sentence. SP then provides agents with optimal learning opportunities by organizing the scene with: 1) the required objects to reach the goal (not too difficult) 2) procedurally-generated distracting objects (not too easy and providing further discovery opportunities). This constitutes a developmental scaffolding modelling the process of Zone of Proximal Development (ZPD) introduced by Vygotsky to describe infant-parent learning dynamics (Vygotsky, 1978).
- At the end of each episode (right of Fig. 8.2), SP utters a set of sentences describing achieved and meaningful outcomes (except sentences from a test set). Linguistic guidance given through descriptions is a key component of how parents "teach" language to infants, which contrasts with instruction following (providing a linguistic command and then a reward), which is rarely seen in real parent-child interactions (Tomasello, 2005; Bornstein et al., 1992). By default, SP respects the 3 following properties: *precision*: descriptions are accurate, *exhaustiveness*: it provides all valid descriptions for each episode and *full-presence*: it is always available. Sec. 8.5.5 investigates relaxations of the last two assumptions.

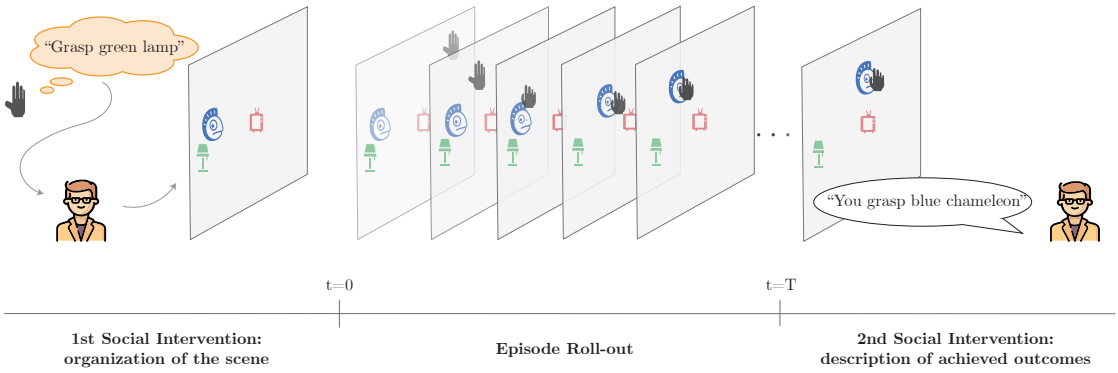


Figure 8.2: **Illustration of the two interventions of SP during an episode.** At the beginning of the episode, SP perceives the agent's goal and organizes the scene with the object required to achieve it as well as distractors. At the end of the episode, SP describes relevant outcomes by uttering a set of sentences.

Pre-verbal infants are known to acquire object-based representations very early (Spelke et al., 1992; Johnson et al., 2003) and, later, to benefit from a simplified parent-child

language during language acquisition (Mintz, 2003). Pursuing a developmental approach (Asada et al., 2009), we assume corresponding object-based representations and a simple grammar. As we aim to design agents that bootstrap creative exploration without prior knowledge of possible interactions or language, we do not consider the use of pre-trained language models.

8.2.2 Simplification of the Playground Environment

We argue that the study of new mechanisms requires the use of controlled environments. We thus propose to conduct our experiments using a simplified version of the playground environment introduced in Sec. 7.2 of chapter 7. More specifically we propose to simplify the agent perception and to consider a simplified grammar for the generation of the descriptions provided by SP. In this new configuration, SP only describes changes in configurations with respect to the initial state and does not provide any past tense or spatial descriptions. We, here, provide details about the agent perception, the behavior of SP and the grammar.

Simplified agent perception and embodiment

Agents have access to state vectors describing the scene: the agent's body and the objects. Each object is represented by a set of features describing its type, position, color, size and whether it is grasped. Categories are not explicitly encoded. Objects are made unique by the procedural generation of their color and size. The agent can perform bounded translations in the 2D plane, grasp and release objects with its gripper. It can make animals and plants grow by bringing them the right supply (food or water for animals, water for plants).

At time step t, we define an observation $\mathbf{o}_t$ as the concatenation of body observations (2D-position, gripper state) and objects' features. These two types of features form affordances between the agent and the objects around. These affordances are necessary to understand the meaning of object interactions like *grasp*. The state $\mathbf{s}_t$ used as input of the models is the concatenation of $\mathbf{o}_t$ and $\mathbf{\Delta o}_t = \mathbf{o}_t - \mathbf{o}_0$ to provide a sense of time. This is required to acquire the understanding and behavior related to the *grow* predicate, as the agent needs to observe and produce a change in the object's size

Grammar

We now present the grammar that generates descriptions for the set of goals achievable in the simplified Playground environment ($\mathcal{G}^A$).

We partition this set of achievable goals into a training ($\mathcal{G}^{\text{train}}$) and a testing ($\mathcal{G}^{\text{test}}$) set. Goals from $\mathcal{G}^{\text{test}}$ are intended to evaluate the ability of our agent to explore the set of achievable outcomes beyond the set of outcomes described by SP. See table D.1 in Supplementary Sec. D.2 for the complete set of testing goals). Note that some goals might be syntactically valid but not achievable. This includes all goals of the form *grow* + ***color*** $\cup$ {*any*} + ***furniture*** $\cup$ {*furniture*} (e.g. *grow red lamp*).

<S>::=	*go* <loc> | *grasp* <all_thing> | *grow* <all_living_thing>
all_thing ::=	<thing> | <attr> <thing>
all_living_thing ::=	<living_thing> | <attr> <living_thing>
<thing> ::=	<object_category> | <living_thing> | <furniture> | <supply>
<living_thing> ::=	<plant> | <animal>
<object_category ::=	*thing* | *living_thing* | *animal* | *plant* | *funiture* | *supply*
<animal> ::=	*dog* | *cat* | *chameleon* | *human* | *fly* | *parrot* | *mouse* | *lion* | *pig* | *cow*
<plant> ::=	*cactus* | *carnivorous* | *flower* | *tree* | *bush* | *grass* | *algae* | *tea* | *rose* | *bonsai*
<furniture> ::=	*door* | *chair* | *desk* | *lamp* | *table* | *cupboard* | *sink* | *window* | *sofa* | *carpet*
<supply> ::=	*water* | *food*
<loc> ::=	*left* | *right* | *top* | *bottom* | *center* | *bottom left* | *bottom right* | *top left* | *top right*
<attr> ::=	*blue* | *green* | *red*

Table 8.1: Updated grammar used by SP do describe the agent's behavior

8.2.3 Evaluation Metrics

This chapter investigates how goal imagination can lead agents to efficiently and creatively explore their environment to discover interesting interactions with objects around them. In this quest, SP guides agents towards a set of interesting outcomes by uttering NL descriptions. Through compositional recombinations of these sentences, goal imagination aims to drive creative exploration, to push agents to discover outcomes beyond the set of outcomes known by SP. We evaluate this desired behavior by three metrics: 1) the generalization of the policy to new states, using goals from the training set that SP knows and describes; 2) the generalization of the policy to new language goals, using goals from the testing set unknown to SP; 3) goal-oriented exploration metrics. These measures assess the quality of the agents' intrinsically motivated exploration. Measures 1) and 2) are also useful to assess the abilities of agents to learn language skills. We measure generalization for each goal as the success rate over 30 episodes and report $\overline{\texttt{SR}}$ the average over goals. We evaluate exploration with the *interesting interaction count* (I2C). I2C is computed on different sets of interesting interactions: behaviors a human could infer as goal-directed. These sets include the training, testing sets and an extra set containing interactions such as bringing water or food to inanimate objects. $\text{I2C}_{\mathcal{I}}$ measures the number of times interactions from $\mathcal{I}$ were observed over the last epoch (600 episodes), whether they were targeted or not (see Supplementary Sec. D.3). Thus, I2C measures the penchant of agents to explore interactions with objects around them. Unless specified otherwise, we provide means μ and standard deviations over 10 seeds and report statistical significance using a two-tail Welch's t-test with null hypothesis $\mu_1 = \mu_2$, at level $\alpha = 0.05$ (noted by star and circle markers in figures) (Colas et al., 2019b).

8.3 The IMAGINE Architecture

IMAGINE agents build a repertoire of goals and train two internal models: 1) a goal-achievement reward function $\mathcal{R}$ to predict whether a given description matches a behavioral trajectory; 2) a policy π to achieve behavioral trajectories matching descriptions. The architecture is presented in Fig. 8.3 and follows this logic:

1. The *Goal Generator* samples a target goal g_{target} from known and imagined goals $(\mathcal{G}_{\text{known}} \cup \mathcal{G}_{\text{im}})$.
2. The agent (*RL Agent*) interacts with the environment using its policy π conditioned on g_{target}.
3. State-action trajectories are stored in a replay buffer $mem(\pi)$.
4. SP's descriptions of the last state are considered as potential goals $\mathcal{G}_{\text{SP}}(\mathbf{s}_T) = \mathcal{D}_{\text{SP}}(\mathbf{s}_T)$.
5. $mem(\mathcal{R})$ stores positive pairs $(\mathbf{s}_T,\ \mathcal{G}_{\text{SP}}(\mathbf{s}_T))$ and infers negative pairs $(\mathbf{s}_T,\ \mathcal{G}_{\text{known}} \setminus \mathcal{G}_{\text{SP}}(\mathbf{s}_T))$.
6. The agent then updates:
   - *Goal Gen.*: $\mathcal{G}_{\text{known}} \leftarrow \mathcal{G}_{\text{known}} \cup \mathcal{G}_{\text{SP}}(\mathbf{s}_T)$ and $\mathcal{G}_{im} \leftarrow \text{Imagination}(\mathcal{G}_{\text{known}})$.
   - *Language Encoder* (L_e) and *Reward Function* $(\mathcal{R})$ are updated using data from $mem(\mathcal{R})$.
   - *RL agent*: We sample a batch of state-action transitions $(\mathbf{s},\ \mathbf{a},\ \mathbf{s}')$ from $mem(\pi)$. Then, we use *Hindsight Replay* and $\mathcal{R}$ to bias the selection of substitute goals to train on (g_s) and compute the associated rewards $(\mathbf{s},\ \mathbf{a},\ \mathbf{s}',\ g_s,\ r)$. Substituted goals g_s can be known or imagined goals. Finally, the policy and critic are trained via RL.

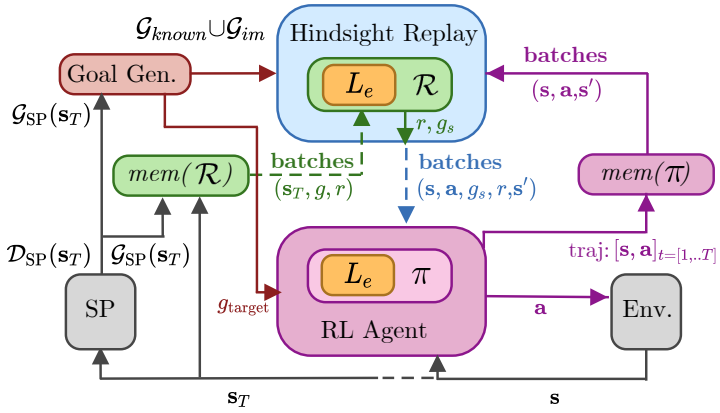


Figure 8.3: **IMAGINE architecture.** Colored boxes show the different modules of IMAGINE. Lines represent update signals (dashed) and function outputs (plain). The language encoder L_e is shared.

Algorithm 4 outlines the pseudo-code of our learning architecture.

Algorithm 4: IMAGINE

Input: env, SP
Initialize: L_e, $\mathcal{R}$, π, $mem(\mathcal{R})$, $mem(\pi)$, $\mathcal{G}_{\text{known}}$, $\mathcal{G}_{\text{im}}$
    # Random initializations for networks
    # empty sets for memories and goal sets
for $e = 1 : N_{episodes}$ **do**
  if $\mathcal{G}_{\text{known}} \neq \emptyset$ **then**
    sample g_{NL} from $\mathcal{G}_{\text{known}} \cup \mathcal{G}_{\text{im}}$
    $g \leftarrow L_e(g_{\text{NL}})$
  else
    sample g from $\mathcal{N}(0, \mathbf{I})$
  end if
  $s_0 \leftarrow$ env.reset()
  for $t = 1 : T$ **do**
    $a_t \leftarrow \pi(s_{t-1}, g)$
    $s_t \leftarrow$ env.step(a_t)
    mem_π.add(s_{t-1}, a_t, s_t)
  end for
  $\mathcal{G}_{\text{SP}} \leftarrow$ SP.get_descriptions(s_T)
  $\mathcal{G}_{\text{known}} \leftarrow \mathcal{G}_{\text{known}} \cup \mathcal{G}_{\text{SP}}$
  $mem(\mathcal{R})$.add(s_T, g_{NL}) for g_{NL} in $\mathcal{G}_{\text{SP}}$
  if goal imagination allowed **then**
    $\mathcal{G}_{\text{im}} \leftarrow$ **Imagination**($\mathcal{G}_{\text{known}}$) # see Algorithm 8
  end if
  Batch$_\pi \leftarrow$ **ModularBatchGenerator**($mem(\pi)$) # Batch$_\pi$=$\{(s, a, s')\}$
  Batch$_\pi \leftarrow$ **Hindsight**(Batch$_\pi, \mathcal{R}, \mathcal{G}_{\text{known}}, \mathcal{G}_{\text{im}}$) # Batch$_\pi$=$\{(s, a, r, g, s')\}$ where $r = \mathcal{R}(s, g)$
  $\pi \leftarrow$**RL_Update**(Batch$_\pi$)
  if e % reward_update_freq == 0 **then**
    Batch$_\mathcal{R} \leftarrow$ ModularBatchGenerator($mem(\mathcal{R})$)
    L_e, $\mathcal{R} \leftarrow$ **LE&RewardFunctionUpdate**(Batch$_\mathcal{R}$)
  end if
end for

8.3.1 Goal Generator

The goal generator is a generative model of NL goals. It generates target goals g_{target} for data collection and substitutes goals g_{s} for hindsight replay. When goal imagination is disabled, the goal generator samples uniformly from the set of known goals $\mathcal{G}_{\text{known}}$, sampling random vectors if empty. When enabled, it samples with equal probability from $\mathcal{G}_{\text{known}}$ and $\mathcal{G}_{\text{im}}$ (set of imagined goals). $\mathcal{G}_{\text{im}}$ is generated using a mechanism grounded in construction grammar that leverages the compositionality of language to imagine new goals from $\mathcal{G}_{\text{known}}$. The heuristic consists in computing sets of *equivalent words*: words that appear in two sentences that only differ by one word. For example, from *grasp red lion* and *grow red lion*, *grasp* and *grow* can be considered *equivalent* and from *grasp green tree* one can imagine a new goal *grow green tree* (see Fig. 8.1f). Imagined goals do not include known goals. Among them, some are meaningless, some are syntactically correct but infeasible (e.g. *grow red lamp*) and some belong to $\mathcal{G}^{\text{test}}$, or even to $\mathcal{G}^{\text{train}}$ before they are encountered by the agent and described by SP. The pseudo-code and all imaginable goals are provided in Supplementary Sec. D.4.

8.3.2 Language Encoder

The language encoder (L_e) embeds NL goals ($L_e : \mathcal{G}^{\text{NL}} \to \mathbb{R}^{100}$) using an LSTM (Hochreiter & Schmidhuber, 1997) trained jointly with the reward function. L_e acts as a goal translator, turning the goal-achievement reward function and policy into language-conditioned functions.

8.3.3 Object-centered Modular Architectures

The goal-achievement reward function, policy and critic leverage novel *modular-attention* (MA) architectures based on Deep Sets (Zaheer et al., 2017), gated attention mechanisms (Chaplot et al., 2018b) and object-centered representations. The idea is to ensure efficient skill transfer between objects, no matter their position in the state vector. This is done through the combined use of a shared neural network that encodes object-specific features and a permutation-invariant function to aggregate the resulting latent encodings. The shared network independently encodes, for each object, an affordance between this object (object observations), the agent (body observations) and its current goal. The goal embedding, generated by L_e, is first cast into an attention vector in $[0,\ 1]$, then fused with the concatenation of object and body features via an Hadamard product (gated-attention (Chaplot et al., 2018b)). The resulting object-specific encodings are aggregated by a permutation-invariant function and mapped to the desired output via a final network (e.g. into actions or action-values). Fig. 8.4 gives an illustration of both the reward and the policy modular architectures.

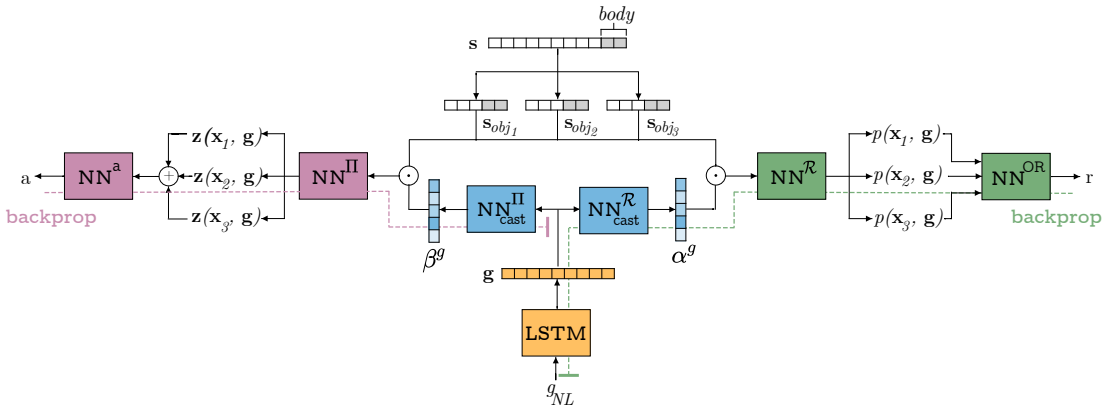


Figure 8.4: **Policy and reward function architectures:** The reward function is represented on the right in green, the policy on the left in pink, the language encoder in the bottom in yellow and the attention mechanisms at the center in blue.

Reward function

Learning a goal-achievement reward function ($\mathcal{R}$) is framed as binary classification: $\mathcal{R}(\mathbf{s}, \mathbf{g}) : \mathcal{S} \times \mathbb{R}^{100} \rightarrow \{0, 1\}$. We use the MA architecture with attention vectors $\boldsymbol{\alpha}^g$, a shared network $\mathrm{NN}^{\mathcal{R}}$ with output size 1 and a logical OR aggregation. $\mathrm{NN}^{\mathcal{R}}$ computes object-dependent rewards r_i in $[0, 1]$ from the object-specific inputs and the goal embedding. The final binary reward is computed by $\mathrm{NN}^{\mathrm{OR}}$ which outputs 1 whenever $\exists j : r_j > 0.5$. We pre-trained a neural-network-based OR function to enable end-to-end training with back-propagation. The overall function is:

$$\mathcal{R}(\mathbf{s}, g) = \mathrm{NN}^{\mathrm{OR}}([\mathrm{NN}^{\mathcal{R}}(\mathbf{s}_{obj(i)} \odot \boldsymbol{\alpha}^g)]_{i \in [1..N]})$$

Data. Interacting with the environment and SP, the agent builds a set of entries $[\mathbf{s}_T, g, r]$ with $g \in \mathcal{G}_{\text{known}}$ where $r \in \{0, 1\}$ rewards the achievement of g in state $\mathbf{s}_T$: $r = 1$ if $g \in \mathcal{G}_{\text{SP}}(\mathbf{s}_T)$ and 0 otherwise. L_e and $\mathcal{R}$ are periodically updated jointly by back-propagation on this dataset.

Multi-goal RL agent

Our agent is controlled by a goal-conditioned policy π (Schaul et al., 2015) based on the MA architecture (see Fig. 8.4). It uses an attention vector $\boldsymbol{\beta}^g$, a shared network NN^{π}, a sum aggregation and a mapper NN^{a} that outputs the actions. Similarly, the critic produces action-values via $\boldsymbol{\gamma}^g$, NN^{Q} and $\mathrm{NN}^{\text{a-v}}$ respectively:

$$\pi(\mathbf{s}, g) = \mathrm{NN}^{\mathrm{a}}\Big(\sum_{i \in [1..N]} \mathrm{NN}^{\pi}(\mathbf{s}_{obj(i)} \odot \boldsymbol{\beta}^g)\Big) \qquad Q(\mathbf{s}, \mathbf{a}, g) = \mathrm{NN}^{\text{a-v}}\Big(\sum_{i \in [1..N]} \mathrm{NN}^{Q}([\mathbf{s}_{obj(i)}, \mathbf{a}] \odot \boldsymbol{\gamma}^g)\Big).$$

Both are trained using DDPG (Lillicrap et al., 2016), although any other off-policy algorithm can be used. As detailed in Supplementary Sec. D.6, our agent uses a form of Hindsight Experience Replay (Andrychowicz et al., 2017a).

8.4 Systematic Generalization

Because systematic generalization to new linguistic goal constructs is a central dimension of the analysis of our proposed IMAGINE algorithm, we provide details about the specific types of generalization we investigate and how the IMAGINE system can achieve them.

8.4.1 Different Types of Generalization

Generalization can occur in two different modules of the IMAGINE architecture: in the reward function and in the policy. Agents can only benefit from goal imagination when their reward function is able to generalize the meanings of imagined goals from the meanings of known ones. When they do, they can further train on imagined goals, which might, in turn, reinforce the generalization of the policy. In a similar fashion as we did in chapter 7 we characterize different types of generalizations that the reward and policy can both demonstrate:

- Type 1 - *Attribute-object generalization*: This is the ability to accurately associate an attribute and an object that were never seen together before. To interpret the goal *grasp red tree* requires to isolate the *red* and *tree* concepts from other sentences and to combine them to recognize a *red tree*. To measure this ability, we removed from the training set all goals containing the following attribute-object combinations: *{blue door, red tree, green dog}* and added them to the testing set (4 goals).
- Type 2 - *Object identification*: This is the ability to identify a new object from its attribute. We left out of the training set all goals containing the word *flower* (4 goals). To interpret the goal *grasp red flower* requires to isolate the concept of *red* and to transpose it to the unknown object *flower*. Note that in the case of *grasp any flower*, the agent cannot rely on the attribute, and must perform some kind of complement reasoning:"if these are known objects, and that is unknown, then if must be a *flower*".
- Type 3 - *Predicate-category generalization*: This is the ability to interpret a predicate for a category when they were never seen together before. As explained in Sec. D.1, a category regroups a set of objects and is not encoded in the object state vector. It is only a linguistic concept. We left out all goals with the *grasp* predicate and the *animal* category (4 goals). To correctly interpret *grasp any animal* requires to identify objects that belong to the animal category (acquired from "growing *animal*" and "growing animal objects" goals), to isolate the concept of *grasping* (acquired from grasping non-*animal* objects) and to combine the two.
- Type 4 - *Predicate-object generalization*: This is the ability to interpret a predicate for an object when they were never seen together before. We leave out all goals with the *grasp* predicate and the *fly* object (4 goals). To correctly interpret *grasp any fly*, the agent should leverage its knowledge about the *grasp* predicate (acquired from the "grasping non-fly objects" goals) and the *fly* object (acquired from the "growing flies" goals).
- Type 5 - *Predicate dynamics generalization*: This is the ability to generalize the behavior associated with a predicate to another category of objects, for which

the dynamics is changed. In the *Playground* environment, the dynamics of *grow* with ***animals*** and ***plants*** is a a bit different. ***animals*** can be grown with *food* and *water* whereas ***plants*** only grow with *water*. We want to see if IMAGINE can learn the dynamics of *grow* on ***animals*** and generalize it to ***plants***. We left out all goals with the *grow* predicate and any of the ***plant*** objects, *plant* and *living thing* categories (48 goals). To interpret, *grow any plant*, the agent should be able to identify the ***plant*** objects (acquired from the "grasping plants" goals) and that objects need supplies (food or water) to *grow* (acquired from the "growing animals" goals). Type 5 is more complex than Type 4 for two reasons: 1) because the dynamics change and 2) because it mixes objects and categories. Note that, by definition, the zero-shot generalization is tested without additional reward signals (before imagination). As a result, even the best zero-shot generalization possible cannot adapt the *grow* behavior from animals to plant and would bring food and water with equal probability $p = 0.5$ for each.

Table D.1 provides the exhaustive list of goals used to test each type of generalization.

8.4.2 Different Ways to Generalize

Agent can generalize to out-of-distribution goals (from any of the 5 categories above) in three different ways:

1. *Policy zero-shot generalization*: The policy can achieve the new goal without any supplementary training.
2. *Reward zero-shot generalization*: The reward can tell whether the goal is achieved or not without any supplementary training.
3. *Policy n-shot generalization or behavioral adaptation*: When allowed to imagine goals, IMAGINE agents can use the zero-shot generalization of their reward function to autonomously train their policy to improve on imagined goals. After such training, the policy might show improved generalization performance compared to its zero-shot abilities. We call this performance *n-shot generalization*. The policy received supplementary training, but did not leverage any external supervision, only the zero-shot generalization of its internal reward function. This is crucial to achieve Type 5 generalization. As we said, zero-shot generalization cannot figure out that plants only grow with water. Fine-tuning the policy based on experience and internal rewards enables agents to perform *behavioral adaptation*: adapting their behavior with respect to imagined goals in an autonomous manner (see Main Fig. 8.5).

8.5 Experiments

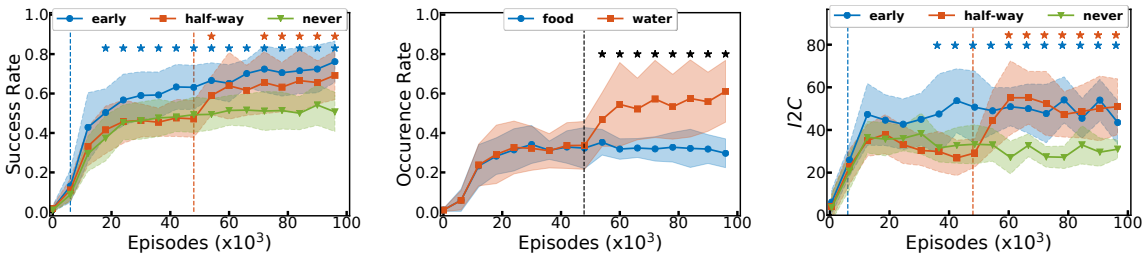
This section first showcases the impact of goal imagination on exploration and generalization (Sec. 8.5.1). For a more complete picture, we analyze other goal imagination mechanisms and investigate the properties enabling these effects (Sec. 8.5.3). Finally, we show that our modular architectures are crucial to a successful goal imagination (Sec. 8.5.4) and discuss more realistic interactions with SP (Sec. 8.5.5). IMAGINE agents achieve near

perfect generalizations to new states (training set of goals): $\overline{\texttt{SR}} = 0.95 \pm 0.05$. We thus focus on language generalization and exploration. Supplementary Sections D.2 to D.7 provide additional results and insights organized by theme (Generalization, Exploration, Goal Imagination, Architectures, Reward Function and Visualizations).

8.5.1 The Impact of Goal Imagination on Generalization and Exploration

Global generalization performance

Fig. 8.5(a) shows $\overline{\texttt{SR}}$ on the set of testing goals, when the agent starts imagining new goals early (after $6 \cdot 10^3$ episodes), half-way (after $48 \cdot 10^3$ episodes) or when not allowed to do so. Imagining goals leads to significant improvements in generalization.



(a) (b) (c)

Figure 8.5: **Goal imagination drives exploration and generalization.** Vertical dashed lines mark the onset of goal imagination. (a) $\overline{\texttt{SR}}$ on testing set. (b) Behavioral adaptation, empirical probabilities that the agent brings supplies to a plant when trying to grow it. (c) I2C computed on the testing set. Stars indicate significance (a and c are tested against *never*).

Behavioral Adaptation

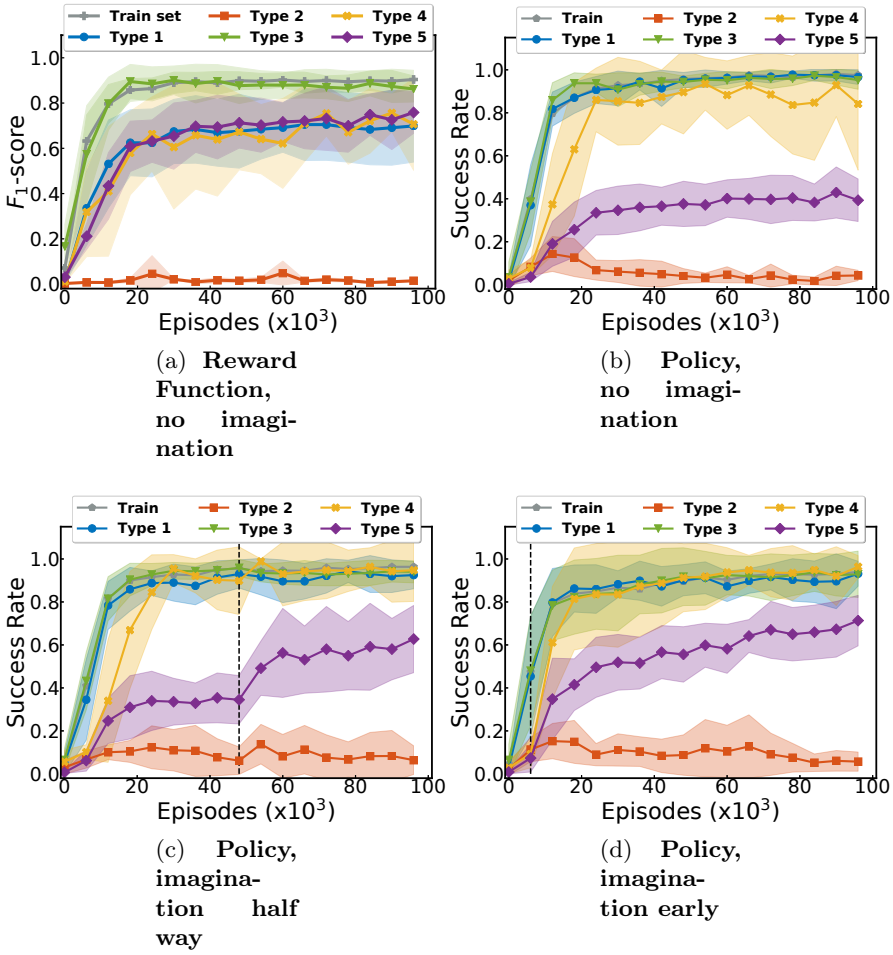
Agents learn to grow animals from SP's descriptions, but are never told they could grow plants. When evaluated offline on the *growing-plants* goals before goal imagination, agents' policies perform a sensible zero-shot generalization and bring them water or food with equal probability, as they would do for animals (Fig. 8.5(b), left). As they start to imagine and target these goals, their behavior adapts (Fig. 8.5(b), right). Because the reward function shows good zero-shot abilities (as demonstrated in Sec. 8.5.2), it only provides positive rewards when the agent brings water. The policy therefore slowly adapts to this internal reward signal and pushes agents to bring more water. We call this phenomenon *behavioral adaptation*.

Exploration

Fig. 8.5(c) presents the I2C metric computed on the set of interactions related to $\mathcal{G}^{\text{test}}$ and demonstrates the exploration boost triggered by goal imagination. Supplementary Sec. D.3 presents other I2C metrics computed on additional interactions sets.

8.5.2 Systematic Generalization

Fig. 8.6 presents training and generalization performance of the reward function and policy. We evaluate the generalization of the reward function via its average F_1 score on $\mathcal{G}^{\text{test}}$, the generalization of the policy by $\overline{\texttt{SR}}_{\text{test}}$.



(a) **Reward Function, no imagination**

(b) **Policy, no imagination**

(c) **Policy, imagination half way**

(d) **Policy, imagination early**

Figure 8.6: **Zero-shot and n-shot generalizations of the reward function and policy.** Each figure represents the training and testing performances (split by generalization type) for the reward (a), and the policy (b, c, d). (a) and (b) represent zero-shot performance in the *no imagination* conditions. In (c) and (d), agents start to imagine goals as denoted by the vertical dashed line. Before that line, $\overline{\texttt{SR}}$ evaluate zero-shot generalization. After, it evaluates the n-shot generalization, as agent can train autonomously on imagined goals.

Reward function zero-shot generalization. When the reward function is trained in parallel of the policy, we monitor its zero-shot generalization capabilities by computing the F_1-score over a dataset collected separately with a trained policy run on goals from $\mathcal{G}^{\text{test}}$ (kept fixed across runs for fair comparisons). As shown in Fig. 8.6a, the reward function exhibits good zero-shot generalization properties over 4 types of generalization after 25×10^3 episodes. Note that, because we test on data collected with a different

RL policy, the F_1-scores presented in Fig. 8.6a may not faithfully describe the true generalization of the reward function during co-training.

Policy zero-shot generalization. The zero-shot performance of the policy is evaluated in Fig. 8.6b (*no imagination* condition) and in the period preceding goal imagination in Fig. 8.6c and 8.6d (before vertical dashed line). The policy shows excellent zero-shot generalization properties for Type 1, 3 and 4, average zero-shot generalization on Type 5 and fails to generalize on Type 2. Type 1, 3 and 4 can be said to have similar levels of difficulty, as they all require to learn two concepts individually before combining them at test time. Type 2 is much more difficult as the meaning of only one word is known. The language encoder indeed receives a new word token which seems to disturb behavior. As said earlier, zero-shot generalization on Type 5 cannot do better than 0.5, as it cannot infer that plants only require water.

Policy n-shot generalization. When goal imagination begins (Figures 8.6c and 8.6d after the vertical line), agents can imagine goals and train on them. This means that $\overline{\texttt{SR}}$ evaluates n-shot policy generalization. Agents can now perform *behavior adaptation*. They can learn that plants need water. As they learn this, their generalization performance on goals from Type 5 increases and goes beyond 0.5. Note that this effects fights the zero-shot generalization. By default, policy and reward function apply zero-shot generalization: e.g. they bring water or food equally to plants. Behavioral adaptation attempts to modify that default behavior. Because of the poor zero-shot generalization of the reward on goals of Type 2, agents cannot hope to learn Type 2 behaviors. Moreover, Type 2 goals cannot be imagined, as the word *flower* is unknown to the agent.

8.5.3 Ablation on Goal Imagination Mechanisms

Properties of imagined goals.

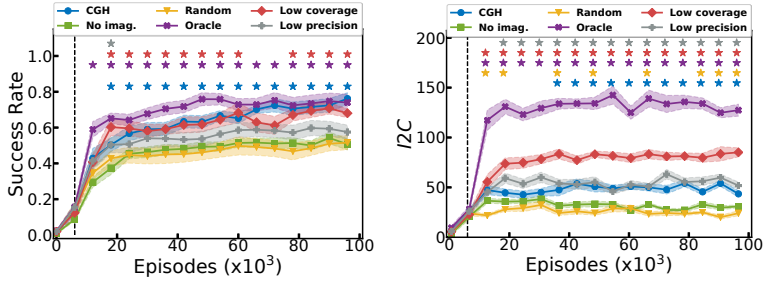
We propose to characterize goal imagination mechanisms by two properties: 1) *Coverage*: the fraction of $\mathcal{G}^{\text{test}}$ found in $\mathcal{G}_{\text{im}}$ and 2) *Precision*: the fraction of the imagined goals that are achievable. We compare our goal imagination mechanism based on the construction grammar heuristic (CGH) to variants characterized by 1) lower coverage; 2) lower precision; 3) perfect coverage and precision (oracle); 4) random goal imagination baseline (random sequences of words from $\mathcal{G}^{\text{train}}$ leading to near null coverage and precision). These measures are computed at the end of experiments, when all goals from $\mathcal{G}^{\text{train}}$ have been discovered (Fig. 8.7a).

Fig. 8.7b shows that CGH achieves a generalization performance on par with the oracle. Reducing the coverage of the goal imagination mechanism still brings significant improvements in generalization. Supplementary Sec. D.4 shows, for the *Low Coverage* condition, that the generalization performance on the testing goals that were imagined is not statistically different from the performance on similar testing goals that could have been imagined but were not. This implies that the generalization for imagined goals also benefits similar non-imagined goals from $\mathcal{G}^{\text{test}}$. Finally, reducing the precision of imagined goals (gray curve) seems to impede generalization (no significant difference with the *no imagination* baseline). Fig. 8.7c shows that all goal imagination heuristics enable a significant exploration boost. The random goal baseline acts as a control condition.

It demonstrates that the generalization boost is not due to a mere effect of network regularization introduced by adding random goals (no significant effect w.r.t. the *no imagination* baseline). In the same spirit, we also ran a control using random goal embeddings, which did not produce any significant effects.

	Cov.	Pre.
CGH	0.87	0.45
Oracle	1	1
Low Cov.	0.44	0.45
Low Pre.	0.87	0.30
Random G.	≈0	≈0

(a)



(b) (c)

Figure 8.7: **Goal imagination properties.** (a) Coverage and precision of different goal imagination heuristics. (b) $\overline{\text{SR}}$ on testing set. (c) I2C on $\mathcal{G}^{\text{test}}$. We report *sem* (standard error of the mean) instead of *std* to improve readability. Stars indicate significant differences w.r.t the *no imagination* condition.

8.5.4 Interactions Between Modularity and Imagination

Table 8.2: Policy architectures performance. $\overline{\text{SR}}_{\text{test}}$ at convergence.

	MA *	FA
Im.	0.76 ± 0.1	0.15 ± 0.05
No Im.	0.51 ± 0.1	0.17 ± 0.04
p-val	4.8e-5	0.66

We compared MA to flat architectures (FA) that consider the whole scene at once. As the use of FA for the reward function showed poor performance on $\mathcal{G}^{\text{train}}$, Table 8.2 only compares the use of MA and FA for the policy. MA shows stronger generalization and is the only architecture allowing an additional boost with goal imagination. Only MA policy architectures can leverage the novel reward signals coming from imagined goals and turn them into *behavioral adaptation*. Supplementary Sec. D.5 provides additional details.

8.5.5 Social Feedback Properties

We study the relaxation of the *full-presence* and *exhaustiveness* assumptions of SP. We first relax *full-presence* while keeping *exhaustiveness* (blue, yellow, and purple curves). When SP has a 10% chance of being present (yellow), imaginative agents show generalization performance on par with the unimaginative agents trained in a full-presence setting (green), see Fig. 8.8). However, when the same amount of feedback is concentrated in

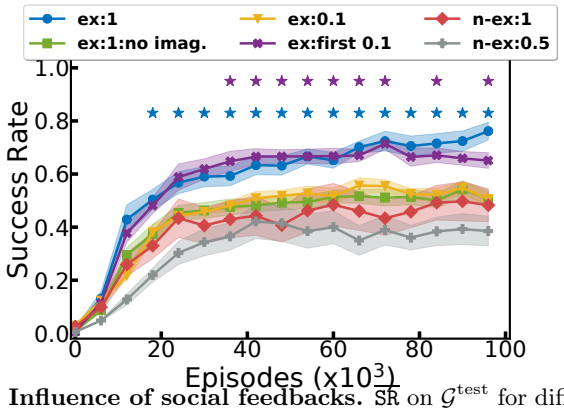


Figure 8.8: **Influence of social feedbacks.** SR on $\mathcal{G}^{\text{test}}$ for different social strategies. Stars indicate significant differences w.r.t. *ex:1 no imag.*. sem plotted, 5 seeds.

the first 10% episodes (purple), goal imagination enables significant improvements in generalization (w.r.t. green). This is reminiscent of children who require less and less attention as they grow into adulthood and is consistent with Chan et al. (2019a). Relaxing *exhaustiveness*, SP only provides one positive and one negative description every episode (red) or in 50% of the episodes (gray). Then, generalization performance matches the one of unimaginative agents in the exhaustive setting (green).

8.6 Discussion and Conclusion

IMAGINE is a learning architecture that enables autonomous learning by leveraging NL interactions with a social partner. As other algorithms from the IMGEP family, IMAGINE sets its own goals and builds behavioral repertoires without external rewards. As such, it is distinct from traditional instruction-following RL agents. This is done through the joint training of a language encoder for goal representation and a goal-achievement reward function to generate internal rewards. Our proposed modular architectures with gated-attention enable efficient out-of-distribution generalization of the reward function and policy. The ability to imagine new goals by composing known ones leads to further improvements over initial generalization abilities and fosters exploration beyond the set of interactions relevant to SP. Our agent even tries to grow pieces of furniture with supplies, a behavior that can echo the way a child may try to feed his doll.

IMAGINE does not need externally-provided rewards but learns which behaviors are *interesting* from language-based interactions with SP. In contrast with hand-crafted reward functions, NL descriptions provide an easy way to guide machines toward relevant interactions. *A posteriori* counterfactual feedback is easier to communicate for humans, especially when possible effects are unknown and, thus, the set of possible instructions is undefined. Hindsight learning also greatly benefits from such counterfactual feedback and improves sample efficiency.

Attention mechanisms further extend the interpretability of the agent's learning by mapping language to attentional scaling factors (see Supplementary Fig. D.10). In addition, Sec. 8.5.5 shows that agents can learn to achieve goals from a relatively small number of descriptions, paving the way toward human-provided descriptions.

Playground is a tool that we hope will enable the community to further study under-explored descriptive setups with rich combinatorial dynamics, as well as goal imagination. It is designed for the study of goal imagination and combinatorial generalization. Compared to existing environments (Hermann et al., 2017a; Chevalier-Boisvert et al., 2019a; Chan et al., 2019a), we allow the use of descriptive feedback, introduce the notion of object categories and category-dependent object interactions (*Grow* refer to different modalities for *plants* or *animals*). Playground can easily be extended by adding objects, attributes, and category- or object-type-dependent dynamics.

IMAGINE could be combined with unsupervised multi-object representation learning algorithms (Burgess et al., 2019; Greff et al., 2019) to work directly from pixels, practically enforcing object-centered representations. The resulting algorithm would still be different from goal-as-state approaches (Nair et al., 2018b; Pong et al., 2020; Nair et al., 2020). Supplementary Sec. D.8 discusses the relevance of comparing IMAGINE to these works. Some tasks involve instruction-based navigation in visual environments that do not explicitly represent objects (Shridhar et al., 2020). Here, also, imagining new instructions from known ones could improve exploration and generalization. Finally, we believe IMAGINE could provide interesting extensions in hierarchical settings, like in Jiang et al. (2019a), with novel goal imagination boosting low-level exploration.

Future work

A more complex language could be introduced, for example, by considering object relationships (e.g. *Grasp any X left of Y*), see (Karch et al., 2020) for a preliminary experiment in this direction. While the use of pre-trained language models (Radford et al., 2019) does not follow our developmental approach, it would be interesting to study how they would interact with goal imagination. Because CGH performs well in our setup with a medium precision (0.45) and because similar mechanisms were successfully used for data augmentation in complex NLP tasks (Andreas, 2020), we believe our goal imagination heuristic could scale to more realistic language.

We could reduce the burden on SP by considering unreliable feedbacks (lower precision), or by conditioning goal generation on the initial scene (e.g. using mechanisms from Cideron et al. (2020c)). One could also add new interaction modalities by letting SP make demonstrations, propose goals or guide the agent's attention. Our modular architectures, because they are set functions, could also directly be used to consider variable numbers of objects. Finally, we could use off-policy learning (Fujimoto et al., 2019) to reinterpret past experiences in the light of new imagined goals without any additional environment interactions.

Links.

Demonstration videos are available at `https://sites.google.com/view/imagine-drl`. The source code of the playground environment can be found at `https://github.com/flowersteam/playground_env` and the source code of the IMAGINE architecture `https://github.com/flowersteam/Imagine`.

Part Summary

Part II of this manuscript started with the presentation of the Vygotskian Autotelic AI framework, which extends the autotelic RL framework presented in chapter 3 to design artificial agents that interact with our rich socio-cultural worlds and internalize pre-existing cultural conventions to become better learners.

We followed this conceptual contribution with two computational contributions focusing on the two categories of internal modules of Vygotksian autotelic agents, the extractive and productive modules.

In chapter 7, we looked at the role of relational inductive biases on the systematic generalization capabilities of an (extractive) language-conditioned reward function. More specifically, we observed that using transformer architectures that maintain object identity is primordial to ground the meaning of complex spatiotemporal concepts describing behavioral trajectories of artificial agents.

In chapter 8, we investigated how language productivity can be used as a cognitive tool to imagine creative goals during curiosity-driven exploration. In our analysis of the IMAGINE system we showed that leveraging construction grammar is an efficient strategy to recombine linguistic constructions into new orders to create goals that are out of the distribution of the effects described by the social partner.

Part III

Discussion

Chapter 9

Summary

Contents

9.1 Summary of our Contributions

The primary objective of this research is to make progress toward designing artificial agents that evolve in sociocultural worlds. Our research contributions are organized around two scientific questions: 1) The formation of cultural conventions in populations of artificial agents and 2) The exploitation of cultural conventions during the cognitive development of artificial agents. These two complementary lines of research are part of the emerging field of developmental AI which integrates traditional AI paradigms (presented in our background chapter) with insights from developmental psychology and linguistics. Both scientific questions are concerned with mechanisms of self-organization: with the first investigating the development of cultural conventions, and the second examining their impact on developmental trajectories. This study investigates the concept of compositionality in both research areas. The first inquiry focuses on how exposure to compositional stimuli can lead to the emergence of compositional conventions, while the second inquiry explores how compositionality can be leveraged for open-ended learning.

9.1.1 Insights from Our Computational Studies on Cultural Convention Formation

The first part of this manuscript tackled the **self-organization of cultural conventions** between artificial agents. It built upon the language formation framework (Sec. 3.2) to investigate the self-organization of graphical sensory-motor communication between

two artificial agents engaged in referential games and the self-organization of goal-directed protocols in the Architect-Builder problem, where agents have asymmetries of information and affordances. These two experimental contributions were presented in chapters 4 and 5, respectively.

The first one introduced CURVES: an algorithm that optimizes a graphical and continuous utterance that names visual referents by relying on learning an energy landscape that represents the alignment between utterances and referents. Through analyzing the performance of CURVES in graphical referential games, we found that contrastive representation learning is an efficient method for agents to self-organize a shared graphical lexicon based on sensory-motor continuous constraints. We also noted that although a pair of agents may show satisfactory compositional generalization performance on a productivity test, this does not necessarily imply that the graphical signs that emerge are compositional. To further understand the nature of the resulting signs and their relationship to compositional language, we proposed to analyze their geometrical structure using the Hausdorff distance but found that basic compositional rules could not explain the compositional generalization.

Our second experimental contribution presented ABIG, which trains an architect and a builder to effectively communicate and solve the Architect-Builder problem. The study of ABIG showed that shared-intent and interaction frames are two relevant priors to facilitate the emergence of a low-level, high-frequency, guiding communication protocol that not only enables an architect-builder pair to solve the task at hand, but that can also generalize to unseen tasks. The fine analysis of ABIG learning dynamics revealed that forgetting mechanisms were instrumental to successful communication. Without it, the builder enters a failure mode and cannot forget wrong action/message associations leading to a non-controllable behavior. Finally, our analysis showed that increasing the number of messages available to the architect (size of vocabulary) is primordial to achieving certain tasks. A vocabulary size equal to the number of actions available to the builder is not enough.

9.1.2 Insights from our Computational Studies on Cultural Convention Exploitation

The second part of this manuscript focused on the **exploitation of pre-existing cultural conventions during the self-organization of developmental trajectories** of artificial agents. We started part II with the introduction of the Vyogtksian Autotelic AI framework (in chapter 6). This framework aims at designing artificial agents that interact with our rich socio-cultural worlds and internalize pre-existing cultural conventions to become better learners. We followed this conceptual contribution with two experimental ones.

In chapter 7, we proposed to equip agents with transformer neural network architectures to enable them to align their experience of the world with linguistic descriptions provided by a surrogate social partner, facilitating systematic generalization. Our rationale for this proposal was based on the observation that grounding the meaning of spatio-temporal concepts is a multi-layered relational problem that involves various interrelated relations such as affordance (subject-object trace), linguistic (world-words),

and spatio-temporal (intra-trace objects) relations. We demonstrated that employing transformer networks led to a significant improvement in systematic generalization performance compared to LSTM baselines. We observed that maintaining object identity in the attention computation of our Transformers is instrumental to achieving good performance on generalization overall and that summarizing object traces in a single token has little influence on performance.

Finally, in chapter 8 we detailed the implementation of IMAGINE: a Vygotskian autotelic agent that converts linguistic descriptions given by a social partner into targetable goals. We showed that IMAGINE can leverage language productivity and systematic generalization to grow an open-ended repertoire of skills in a creative way. In our analysis of the IMAGINE system we showed that construction grammar was an efficient strategy to produce novel goals that are out of the distribution of the effects described by the social partner. We furthermore, demonstrated that modularity and object-centered architectures were instrumental to reaching those invented and novel goals. Crucially, we identified that grounding the meaning of descriptions in observed outcomes, through the learning of a reward function that predicts the compatibility between observations and descriptions, enables agents to perform behavioral adaptation: a correction of overgeneralization to exception in the compositional dynamics of the playground environment (plants only requiring water to grow should not be given food).

9.2 An Alternative Way to Read this Manuscript

The previous section summarized our contributions in a linear and systemic organization: we first explore how artificial agents can self-organize cultural conventions from tabula-rasa and then assume pre-existing conventions to investigate how they can impact skill acquisition. This linear progression ended with the presentation of the IMAGINE agent that leverages cultural conventions to creatively explore its environment. In this paragraph, we argue that this manuscript can also be read backward. Starting from the observation that cultural conventions are a necessary condition for the design of open-ended learners, understanding the formation of cultural conventions becomes of primordial importance. It can, for instance, help develop new learning scenarios for agents such as the Architect Builder problem, and new inductive biases for learning architectures such as the multi-modal transformers that we presented in chapter 7. As a matter of fact, the contributions constituting this research were presented in anti-chronological order. My research journey started with the development of the IMAGINE agent and ended so far with the introduction of CURVES.

9.3 Open-source Code

The entirety of this research builds on open-source code. The specifics of the algorithms and environments created for each project are presented in table 9.1. Each repository contains guidelines to replicate all experiments presented in this manuscript, ensuring that future researchers can readily utilize our computational contributions.

Repo	URL	Chapter	Details
graphical-referential-game	https://github.com/flowersteam/graphical-referential-game	4	Graphical referential game implementation with a sensory-motor system relying on a differentiable sketching library. CURVES training and testing scripts.
architect-builder-abig	https://github.com/flowersteam/architect-builder-abig	5	Training and testing scripts for the ABIG algorithm
architect-builder-env	https://github.com/flowersteam/architect-builder-env	5	Code for the Architect-Builder environment containing a grid world, a communication channel, and an observation blender to combine both observations.
playground	https://github.com/flowersteam/playground_env	7 & 8	Playground environment implemented in pygame. The repository contains two branches: `main` branch is the vanilla version used to experiment with IMAGINE, `temporal_descr` is the version used to create the dataset of our spatio-temporal language grounding experiments.
spatio-temporal-language-transformers	https://github.com/flowersteam/spatio-temporal-language-transformers	7	Links to the datasets used for the experiments on the temporal extension of Playground as well as the multi-modal transformer architecture definitions and training scripts.
imagine	https://github.com/flowersteam/Imagine	8	Agent modules, training and testing scripts of the IMAGINE algorihtm.

Table 9.1: **Summary of open-source code repositories.**

Chapter 10

Perspectives

Contents

This chapter provides future avenues for the integration of cultural convention in AI research. By acknowledging the current limitations of both language evolution setups and Vygotskian autotelic agents we outline several challenges for more human-like, interactive, and culturally informed artificial agents.

10.1 Towards Realistic Models of the Cultural Niche

10.1.1 Scaling Current Neural Network Communicating Agents

In a recent contribution, Chaabouni et al. (2022) posit that "*from a machine learning view, language evolution is deemed as a promising direction to shape agents' representation and design interactive AI*". Despite previous efforts by AI researchers to study language games with neural network agents, the authors concede that recent language evolution experiments have not shown substantial progress over the past two decades and contend that scaling up such experiments is necessary to achieve this objective. As a result, the authors propose to focus on scaling up three central aspects of language games. Specifically, they suggest augmenting the complexity of visual referents by employing more realistic datasets, expanding the number of distractor referents, and considering a

broader population of agents. In this section, we will use these three aspects as a starting point to discuss how current language (or referential) games can be scaled up and made more realistic.

More Realistic Referents. Most of the current approaches only consider either unambiguous referent representations with a low dimensional state space (one-hot vectors) or visual inputs with very few categories and samples. By increasing the number and complexity of referents one may hope to see the emergence of more complex and realistic communication. However, dealing with realistic referents comes with evaluation difficulties. The underlying structure of referents is not known in advance which prevents us from carrying out systematic analyses. Future research may thus look into finding benchmarks for evaluating the structure of emergent conventions in realistic referential games. Such benchmarks may require humans to intervene and provide feedback on the quality of the emerging protocols. Alternatively, experiments could be carried out in a hybrid format with humans interacting with artificial agents just like in the Talking Heads experiment (Steels, 2015) but with modern AI technics and more complex mechanisms for representation learning. The involvement of humans would influence the evolution of agents' representation and thus affect the structure of the emergent protocols. Beyond the augmentation of current visual referents, future work in language games may look at multimodal referents such as videos or sounds (Arandjelovic & Zisserman, 2017).

More Realistic Communication Channel. As mentioned in the motivations of chapter 4, most of the current approaches to language games assume pre-defined discrete channels of communication. This comes in contrast with human communication, which instead relies on a sensory-motor channel, where motor commands produced by the speaker (e.g. vocal or gestural articulators) result in sensory effects perceived by the listener (e.g. audio or visual). With the introduction of the GREG, we integrated sensory-motor constraints in the communication channel between agents to investigate the emergence of a discrete lexicon from continuous graphical utterances. But our contribution is only the first step towards modeling language formation in more ecological settings. Future work may, for instance, explore the emergence of communication in other configurations with different sensory-motor apparatus. The continuous medium in which communication might emerge provides exciting opportunities for analyzing the structure of emergent lexicons. Instead of relying on the standard topographic similarity measure (Brighton & Kirby, 2006), which measures the Spearman correlation between the pairwise distances in the input and message spaces, we can exploit the sensory-motor modality to come up with more interpretable metrics as illustrated by our geometrical analysis of the emerging lexicon provided in chapter 4.

Another limitation of current approaches to referential games comes from the monodirectional nature of the communication channel. An interesting project would be to investigate a symmetrical channel. Indeed, human communication frequently involves ambiguities, and dialogues present an occasion to alleviate this issue. Allowing the speaker to request clarification could potentially generate novel forms of communication.

More Realistic Population of Agents. If some recent works such as the study of Rodríguez Luna et al. (2020) investigate the impact of internal modules of agents on emergent communications, very few works look at the impact of cognitive architectures on communications. Endowing agents with varying cognitive abilities such as different

memory, perceptual, and sensory-motor constraints is therefore a promising research avenue. Crucially, considering populations of such diverse agents and integrating genetic interactions between them could provide valuable insights into the selective advantage that certain inductive biases may provide for the emergence of communication.

10.1.2 Moving Beyond Traditional Language Games

As outlined in the introduction, humans use language to do far more than name objects. They use it to teach, collaborate, and more generally to take part in the sociocultural world in which they are immersed. Although the exact conditions, purpose, and timeline of language's emergence remain unresolved, it is clear that language originated in a physical world similar to ours, among goal-directed embodied agents performing actions over extended periods. As such, incorporating physical interactions within environments is a promising avenue to investigate the emergence of time-extended and more realistic forms of communication.

In this context, MARL seems to be an appropriate framework. Several works already started to integrate communication channels between situated agents performing collaborative navigation tasks (Niu et al., 2021; Du et al., 2021) and mixed cooperative-competitive tasks (Lowe et al., 2017). However, such studies do not focus on the self-organization of language and instead use pre-defined non-realistic channels that often propagate gradients between agents. This experimental design fundamentally breaks the assumptions of the language formation framework. In these kinds of approaches, communication is viewed as a means for exchanging information rather than negotiating meanings (the communication channel can be seen as a bottleneck layer in a single-agent architecture). Furthermore, these approaches primarily evaluate agents' emerging behavior in terms of task performance by measuring the collected rewards in the environment and do not conduct extensive analyses of communication. Finally, the presence of a centralized reward signal accessible to all agents during training renders MARL systems not necessarily ideal for modeling the emergence of communication between independent agents. In contrast, recent work by Kalinowska et al. (2022) specifically investigates the role of multi-step interactions in the emergence of communication. They consider a collaborative navigation environment and showed that agents can learn protocols that enable them to solve tasks. Crucially, they show that memory mechanisms provide flexibility around message timing and lead to the emergence of novel and more abstract meanings.

Building upon the situated emergence of communication, several longer-term research directions can be explored. With the aim to see the emergence of more abstract forms of communication, one could for instance imagine extending the autotelic framework to consider several agents concurrently learning to represent, pursue and communicate about goals (recent work by Masquil et al. (2022) provide the first steps in this direction). By increasing the complexity of the physical world within which agents are situated, one can also hope to implement agent-based models to test various hypotheses regarding the ecological factors implicated in language evolution. A possible experiment could explore the role of communication in simulated hunting games to verify the two-step theory according to which humans started using very simple signed communication during hunting before complexifying them to respond to the increased demand to coordinate

group-hunting efforts (Számadó, 2010). Another promising experiment would be to look into the role of the discovery of fire in the evolution of language. According to Wiessner (2014), the mastery of fire and, in particular, the ability to maintain firelight during the night significantly extended the length of the day, thereby creating new opportunities for humans to communicate, which in turn led to the emergence of firelight talks.

10.1.3 Toward the Formation of Artificial-Cultural Niches

According to Boyd, humans became successful at adapting to the wide variety of environments across the globe thanks to their "*uniquely developed ability to learn from others*". (Boyd et al., 2011). This ability to learn from others was honed through the evolution of human culture, which was largely driven by the emergence of language. Specifically, Boyd argues that:

> "*First language, then narrative, then fiction, created niches that altered selection pressures and made us ever more deeply dependent on knowing more about our kind and our risks and opportunities than we could discover through direct experience.*"(Boyd, 2018)

This argument resonates with the claim of Chaabouni et al. (2022) that language evolution is a promising area of study for making progress toward designing competent artificial agents. While current AI research has primarily focused on the language niche, modeling the formation of more advanced niches, such as storytelling, could lead to significant progress. These niches are characterized by diverse socio-cultural artifacts that humans rely on to learn, memorize, and transmit knowledge. Future AI research may examine the formation of knowledge repertoires, maps, educational tools, and other technologies and instruments. However, it is challenging to recreate the conditions that facilitated human evolution and scale such processes to cover the thousands of years it took humans to develop these abilities. Instead, a more practical approach might be to embed current artificial agents in our rich socio-cultural world to allow them to learn and exploit the cultural niche and artifacts we have already constructed. These ideas are explored further in the next section.

10.2 Towards Vygotksian Autotelic Agents

In order to let agents benefit from our cultural artifacts and to design truly Vygotskian autotelic agents, i.e. agents that extract both language content and structure in order to leverage it to transform their cognitive abilities, we identify four main challenges.

10.2.1 Immersing Autotelic Agents in Rich Socio-Cultural Worlds

To benefit from language, Vygotskian autotelic agents must be immersed into rich socio-cultural worlds close to ours. This will require progress along two dimensions: 1) increasing the richness of their world and 2) augmenting their interactivity and teachability.

What do we mean by *rich* worlds? One aspect is the multimodality of perceptions. Beyond its linguistic dimension, culture is indeed multimodal. Socio-cultural interactions are not always linguistic but often non-verbal as they may involve motor, perceptual or emotional dimensions. The second aspect is socio-cultural situatedness: autotelic agents must interact with other agents and with humans. Scaling the richness of these worlds may thus require the involvement of the video game industry, and specialists in complex, realistic multimodal worlds. Human-in-the-loop research will also be required to let humans enter these rich virtual worlds, for instance via virtual reality technology.

Vygotskian autotelic agents will need to be more interactive and teachable. In a recent paper, Sigaud and colleagues discuss this challenge through a detailed analysis of children's learning abilities and teacher-child interactions (Sigaud et al., 2021). They present a checklist of properties that future Vygotskian autotelic agents must demonstrate to be considered *teachable*. To interact with humans, Vygotskian autotelic agents will also need to target goals in multiple modalities (e.g. linguistic, perceptual, emotional) with various levels of abstraction (Colas et al., 2022b). Modular autotelic architectures may be used to that end. By handling multiple goal spaces in parallel, they can leverage cross-domain hindsight learning: using experience collected while aiming at a goal to learn about other goals in other domains (Colas et al., 2019a).

10.2.2 Enabling Artificial Mental Life with Systematic Internalized Language Production

Only a few approaches internalize language production within agents. So far limited to a few use cases, language production should concern every possible linguistic feedback agents could receive: instructions, corrections, advice, explanations, or cultural artefacts. This internal language production is akin to an artificial *inner speech*, the embryo of *artificial mental life*. Looping back to the constitutive thesis of Carruthers presented in chapter 6, inner speech acts as a common currency for inner modules to exchange information (see a recent implementation of this idea in Zeng et al. (2022)). Combined with world models, inner speech could trigger the simulation of perceptual experience (images, sounds), sensorimotor trajectories, the imagination of possible futures or past memories. Observing these hallucinations, agents could produce new behaviors and new inner speech. This inner loop acts as a mental life that could help agents reason; trigger memories or mnemotechnic representations acting as cognitive aids. As noted by Dove, this account is fully compatible with the embodied hypothesis in cognitive science (Dove, 2018). Following this hypothesis, thinking and modeling sensorimotor experience are one and the same. Here, language brings another set of inputs and outputs for these simulation models and the simulation of abstract content (words, analogical structures, etc) might offer us the capacity to reason abstractly.

10.2.3 Building Editable and Shareable Cultural Models with Aligned LLMs

Large language models encode a lot of information about the human cultures that generated the texts they were trained on LLMs (West et al., 2022; Schramowski et al., 2022).

They can be viewed as (partial) cultural models: by tapping into them, agents could learn about human cultures. They could learn about foundational human concepts, causality, folk psychology, politeness, ethics and all these physical or cultural information that are the subject of everyday stories: fiction, news, or even simple narratives parents use to explain everyday things to children.

Such proxies to human cultures provide both opportunities and challenges. Using existing LLMs as is, we could for example prompt them to generate new goals for exploration or even full curricula based on descriptions of the agent's current abilities and environmental descriptions. We could use LLMs to predict the outcome of the agent's actions given the context and use this to plan in abstract search spaces. We could let agents ask LLMs for guidelines only when they cannot solve the problem themselves (active learning), and more generally to augment the world state with commonsense knowledge.

But letting autotelic agents rely on LLMs might also bring some downsides. Cultural information, because it biases the search space, may limit exploration and lead to the premature abandonment of promising avenues (Bonawitz et al., 2011) (e.g. in astronomy, the cultural support for the geocentric model significantly delayed the acceptance of the heliocentric model). LLMs are also known to convey false information and harmful biases, either because they inadequately learned to encode a culture or because they were trained on cultural artefacts which contained such biases (Shah et al., 2020; Liang et al., 2021; Weidinger et al., 2021; Bender et al., 2021). Autotelic agents relying on these models could demonstrate harmful behaviors and contribute to reinforcing stereotypes and inequalities. The use of LLMs will thus require advances in bias mitigation strategies (Liang et al., 2021; Bender et al., 2021) and improved alignment methods to make LLMs more reliable, trustworthy and moral. Ideally, we want them to model the natural culture agents will be embedded in with high fidelity and align well with its objectives.

Humans are also biased by the cultural environment they are in. During their education, children are taught to think for themselves and to think critically. Autotelic agents should be taught in the same way. Because they are autonomous embodied machines, they can conduct experiments in the world and empirically test the information they were provided. This physical embodiment is often described as the missing piece for LLMs to truly understand the world (Bisk et al., 2020). To address this limitation, recent work by Carta et al. (2023) proposes using functional grounding to align the knowledge captured by LLMs with the environment. This approach involves updating LLMs as they interact with the world using online reinforcement learning.

Just like human cultural narratives can be shifted by government, policies, advertising, activism, and pop culture, artificial cultural narratives should become more malleable. Autotelic agents must be given the possibility to steer, edit and extend their cultural models (i.e. LLMs) in light of their embodied experience; to share it and negotiate it with others; i.e. to participate in a shared cultural evolution. Reversely, humans that train language/cultural models should pay great care to build and understand the cultural input they provide, just like they pay great care to the education of their children.

Through this process, agents could learn some of the uniquely human social features described by Tomasello in his recent book (Tomasello, 2019): cooperative thinking, moral identity or social norms. However, a high-quality alignment may require humans to enter the interaction loop, enabling autotelic agents to ground, verify and correct the cultural

knowledge they acquired with LLMs. Furthermore, mere exposure to culture (either through LLMs or direct interaction with humans) might not be sufficient to design truly autonomous social learners. This may require encoding certain mechanisms such as joint intentionality or other collaborative priors inside agents — the topic of social RL (Jaques et al., 2019).

10.2.4 Pursuing Long-term Goals

Current autotelic agents mostly pursue goals at the timescale of an episode. Humans, on the other hand, can pursue goals they can barely hope to achieve within their lifetime (e.g. building an efficient fusion reactor). Because there is an infinity of potential goals and little time to explore them alone, autotelic agents may need cultural models to bias their selection of long-term goals towards more feasible, interesting, or valuable options — turning an individual exploration into a *population-based exploration*. Keeping long-term goals in mind will require improvements in architecture's memory systems, but might also benefit from language and culture. Indeed, verbalization is known to increase humans' memory span (Elliott et al., 2021) and writing let us set our goals in stone (from the post-it note reminding you to take the garbage out to the Ten Commandments). Young children progressively become future-oriented as they are taught to project themselves into the future through education, social interactions (*what do you want to do when you grow up?*) and cultural metaphors (e.g. the self-made man) (Atance, 2008). If autotelic agents will need better hierarchical RL algorithms to achieve long-term goals, they could also leverage cultural artifacts evolved for improved collaboration and long-term planning — think of roadmaps, organization systems and project management tools (Clark, 1998). Because long-term goals are not rewarding, human cultures supply short-term social rewards (good grades in the educational system, money and social recognition in professional careers) — a form of reward shaping.

Conclusion

As I write the final words, I hope that the brave readers who have reached this point of the manuscript are now convinced that the integration of socio-cultural interactions is a key step towards developing more human-like artificial agents. For my part, I am convinced that there are exciting projects to carry out on this topic. These projects will have the long-term goal of finding the self-organizing attractors within the vast landscape of agents' socio-cultural trajectories, ultimately enabling them to achieve a greater degree of general competence. Given that the most sophisticated skills and abstract objectives of humans have cultural roots – Why shouldn't we let agents define their own objectives through exposure to rich sociocultural stimuli? It is important to note, however, that this perspective requires the establishment of ethical and safety regulations to prevent potentially detrimental outcomes.

In a reciprocal fashion, the study of socio-cultural conventions in the development of artificial agents can stimulate interest in interdisciplinary research, enabling a better understanding of the sociocultural trajectory of our species and societies. In essence, science can be seen as the outcome of cultural evolution (Harari, 2014; Moulin-Frier, 2022), an open-ended exploration process that takes place through collaboration and coordination between people with the aim of making creative discoveries. This exploration process is iterative and constantly refined over time: the outcomes of scientific exploration lead to an increased understanding of our environment (via physics), social interactions (via social science), learning capabilities (via cognitive science), or cultural heritage (via history), which in turn provide us with better tools for future discoveries.

Appendices

Appendix A

CURVES

Supplementary Material

This Supplementary Material provides additional derivations, implementation details and results. More specifically:

- Section A provides supplementary implementation details in the form of:
  - Images of testing set of visual referents;
  - Topographic score derivation;
  - Training procedures and hyperparameters;
  - Pseudo-code.
- Section B provides supplementary results:
  - Auto-comprehension generalization performances;
  - Additional Lexicons;
  - Utterances examples across perspectives illustrating coherence;
  - Topographic maps & scores;
  - Composition matrix examples;
  - T-SNEs of embeddings;

A.1 Supplementary Methods

A.1.1 Sensory-Motor System

Dynamic Motion Primitives. This subsection provides additional details about the implementation of the Dynamical Movement Primitives use to produce 2-dimensional trajectories. Our drawing system consists of a 2-dimensional system that mimics the motion of a pen in a plan. Each of the x and y positions of the pen is controlled by a DMP starting at the center of the image and parameterized by 10 weights. These weights are the parameters of the motion of a one-dimensional oscillator that generates a smooth trajectory of 10 points. The parameters of the two DMPs are given in table A.1.

Sketching Library. Trajectories obtained with the DMPs are then mapped to a 52x52 grid which is converted to an image with the `raster` and `softor` functions of the sketching library Mihai & Hare (2021a). The drawing thickness parameter is fixed to $1e-2$.

Parameter	Value
Number of weights	10
Delta time	0.1
Number of points	10
Weights range	[-500, 500]
Position Init.	0

Table A.1: DMP parameters for each of the two coordinate motions

A.1.2 Testing Set

Fig. A.1 displays examples of compositional referents made of 2 features.

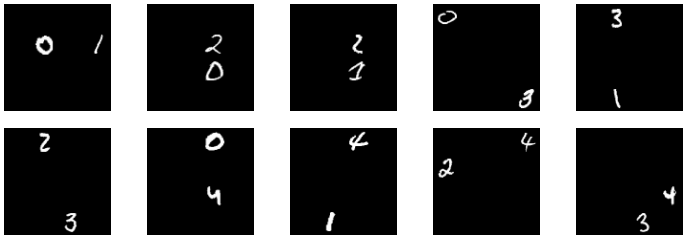


Figure A.1: **Perspective instances of the testing set $\mathcal{R}_5^2$.**

A.1.3 Topographic Score

To evaluate the compositionality of the emerging language we define the topographic score:

$$\rho_{ij} = ||(O, h_{ij})||_2 - ||(O, h_k)||_2 \text{ with } k = \text{argmin}_{k \in \{i,j\}} ||h_k, h_{ij}||_2) \tag{A.1}$$

It is obtained by computing the Hausdorff distance between the utterances denoting compositional referents with respect to both the utterance denoting the single feature i $(d_H(u(r_i), \cdot))$and the one denoting the single feature j $(d_H(u(r_j), \cdot))$. To derive our metric, we define 4 groups of utterances denoting compositional referents.

- $u(r_{ij})$ the utterances for referent made of feature i and j.
- $u(r_{xj}, x \neq i)$ the utterances denoting referent made by composing feature j with any other feature different than i
- $u(r_{iy}, y \neq j)$ the utterances denoting referent made by composing feature i with any other feature different than j
- $u(r_{xy})$ the utterance denoting all other compositional referents in $\mathcal{R}_5^2$.

and compute their Hausdorff distances to $u(r_i)$ and $u(r_j)$. As displayed in Fig. A.2, if utterances $u(r_{ij})$ are compositional we expect them to be at the same time close to $u(r_i)$ and close to $u(r_j)$ and hence to land in the bottom left corner of the distance graph. Moreover, they should be closer to the origin than $u(r_{xj})$ and $u(r_{iy})$. To quantify to what extent it is the case we compute the barycenter of each group h_i, h_j, h_{ij} and h_{xy}

and compute "how closer to the origin" is the compositional barycenter h_{ij} compared to its closest barycenter using equation A.1.

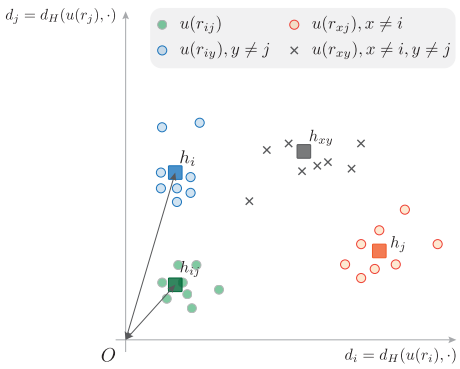


Figure A.2: Idealized mapping of utterances denoting compositional referents in the plan representing distances to utterances naming isolated features i and j.

A.1.4 Training Procedure and Hyperparameters

Agents have two separate encoders based on the same model architecture described in Tab. A.2. Each agent performs association updates with a single step of gradient descent, using its own Adam optimizer with a learning rate of $1e^{-4}$. To allow faster convergence, agents perform an association update between an abstract referent $r_A^\star$ and an utterance u by using a batch of 64 perspectives $\{\Phi(r_A^\star)\}_{i\in[1,64]}$. From a cognitive science perspective, this is comparable to an agent "walking around" an object to better understand how different perceptions relate to the same object. From a computer science perspective, this is similar to the self-supervised framework of SimCLR (Chen et al., 2020), where agents learn representation by contrastively aligning the embeddings of an input with these of the same transformed input.

Layer	Activation
Conv2D(filters=8, stride=2, padding=1)	ReLU
Conv2D(filters=16, stride=2, padding=1)	ReLU
Conv2D(filters=32, stride=2, padding=0)	ReLU
Linear(128)	ReLU
Linear(32)	None

Table A.2: **Model architecture used for both the referent and utterance Encoders.** (when referents are one-hot vectors, the 3 Conv2D layers are replaced by a Linear layer with ReLu activation)

While the drawing pipeline is fully differentiable, it is highly sensitive to local minima. Thus, we solve equation 4.5 in the descriptive case or equation 4.6 in the discriminative scenario by simultaneously performing gradient descent on a batch of 64 randomly initialized command vectors over 100 iterations, using a newly initialized Adam optimizer each time with a learning rate of $1e^{-2}$.

A.1.5 Pseudo-code

Algorithm 5: Speaker's Utterances

Require: perceived referents $\tilde{R}_S$, speaker's referent encoder f_S, speaker's utterance encoder g_S, sensory-motor system M
$Z_r \leftarrow f_S(\tilde{R}_S)$
$c \sim \text{Uniform}()$
for i in range($N_{\text{production}}$) **do**
  $U_S \leftarrow M(c)$
  $Z_u \leftarrow g_S(U)$
  $S \leftarrow \text{sim}_{\cos}(Z_r, Z_u)$
  $\mathcal{L} \leftarrow \text{mean}(\text{diag(S)}) * (-1)$
  GD step on c to minimize $\mathcal{L}$
end for
Return $M(c)$

Algorithm 6: Listener's Selections & Binary Outcomes

Require: perceived referents $\tilde{R}_L$, produced utterances U_S, listener's referent encoder f_L, listener's utterance encoder g_L
$Z_r \leftarrow f_L(\tilde{R}_L)$, $Z_u \leftarrow g_L(U_S)$
$S \leftarrow \text{sim}_{\cos}(Z_r, Z_u)$
$t \leftarrow \text{argmax}(S, \text{axis=1})$
$o \leftarrow \mathbf{0}$
for i in range($N_{\text{referents}}$) **do**
  $o_i \leftarrow \mathbb{1}_{[t_i=i]}$
end for
Return o

Algorithm 7: Agents's Association Losses

Require: perceived referents $\tilde{R}_A$, produced utterances U_A, outcomes o, agent's referent encoder f_A, agent's utterance encoder g_A
$Z_r \leftarrow f_A(\tilde{R}_A)$, $Z_u \leftarrow g_A(U_A)$
$S \leftarrow \text{sim}_{\cos}(Z_r, Z_u)$
$\mathcal{L}_0 \leftarrow CE(S, \text{reduction=False})$, $\mathcal{L}_1 \leftarrow CE(S^\top, \text{reduction=False})$
$\mathcal{L} \leftarrow (\mathcal{L}_0 + \mathcal{L}_1)/2$
if $A =$ "S" **then**
  $\mathcal{L} \leftarrow (\mathcal{L} \cdot o)/N_{\text{referents}}$
else
  $\mathcal{L} \leftarrow (\mathcal{L} \cdot \mathbf{1})/N_{\text{referents}}$
end if
Return $\mathcal{L}$

A.2 Supplementary Results

A.2.1 Auto-comprehension Generalization Performances

Ref.	Auto	Social
One-hot	0.997 ± 0.005	0.991 ± 0.015
Visual-shared	0.862 ± 0.034	0.559 ± 0.027
Visual-unshared	0.425 ± 0.016	0.388 ± 0.02

Table A.3: Descriptive Success Rate

Ref.	Auto	Social
One-hot	0.997 ± 0.005	0.992 ± 0.009
Visual-shared	0.812 ± 0.019	0.567 ± 0.034
Visual-unshared	0.466 ± 0.019	0.404 ± 0.019

Table A.4: Descriminative Success Rate

We define the **Auto** performance metric as the communicative success rate, on test set, for language games involving a single agent playing as both the speaker and listener. We compare **Auto** and **Social** performances (the latter involving pairs of different agents, as done until now) in Tables A.3 & A.4.

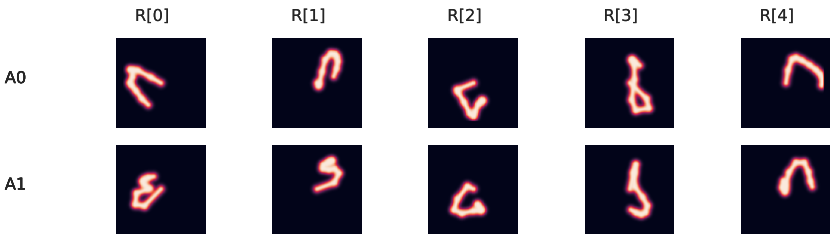


Figure A.3: **Instance of an emerging lexicon.** (Utterances are naming visual-shared referents).

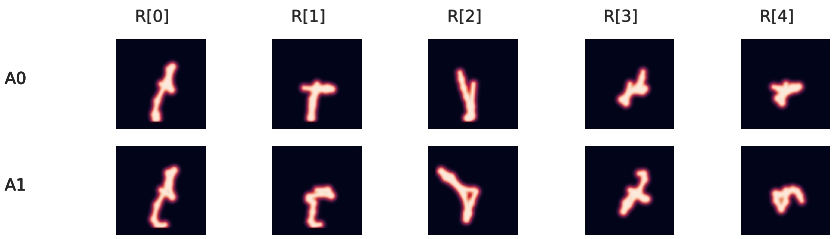


Figure A.4: **Instance of an emerging lexicon.** (Utterances are naming one-hot referents).

A.2.3 Utterances Examples Across Perspectives Illustrating Coherence

The following figures illustrate the P-coherence and A-coherence of an emerging lexicon (Visual-unshared) by displaying, for each referent in R_1, the descriptive utterance produced for 10 random perspectives.

Figure A.5: Utterances examples for referent 0.

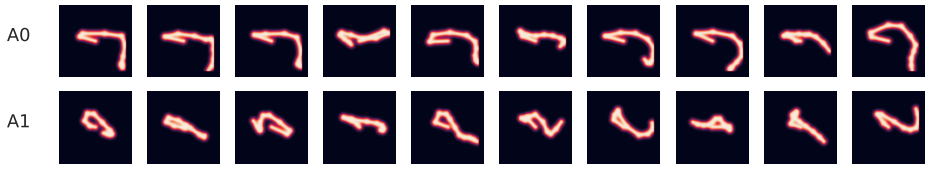


Figure A.6: Utterances examples for referent 1.

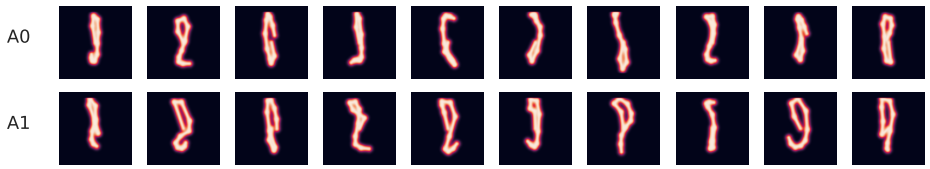


Figure A.7: Utterances examples for referent 2.

Figure A.8: Utterances examples for referent 3.

Figure A.9: Utterances examples for referent 4.

A.2.4 Topographic Maps & Scores

One-Hot

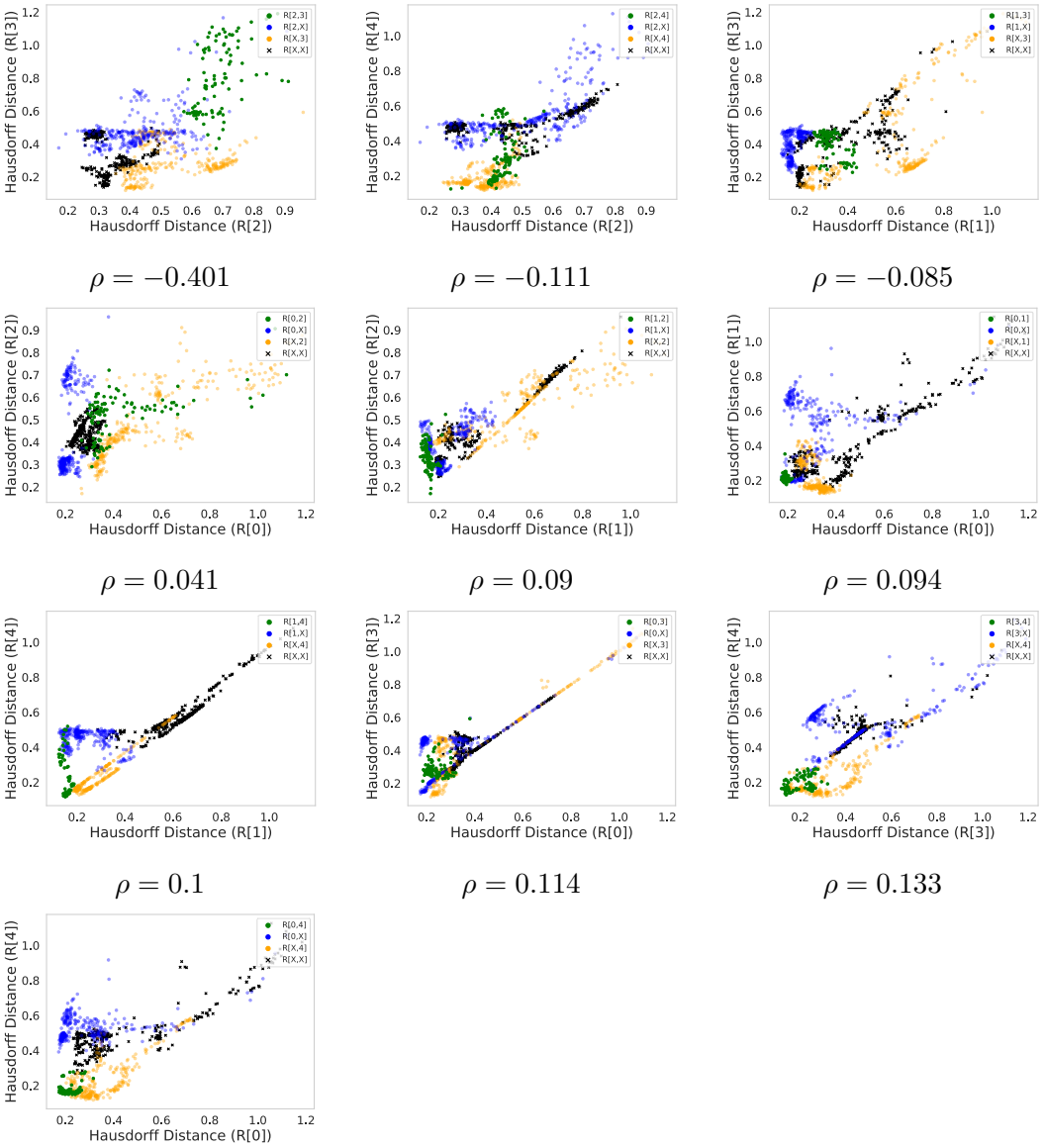


Figure A.10: Topographic maps and their associated topographic scores for each combination of features with one-hot referents

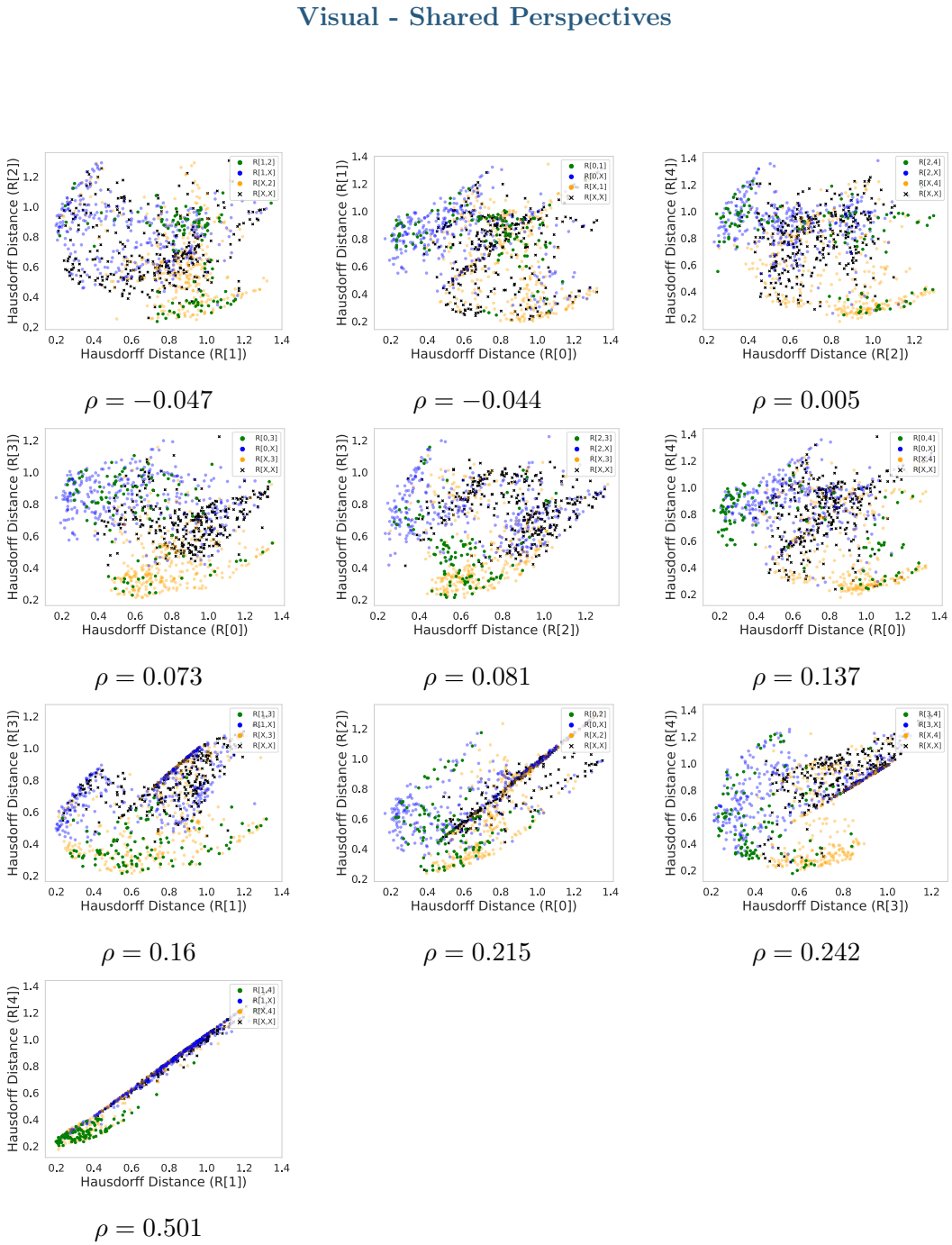


Figure A.11: Topographic maps and their associated topographic scores for each combination of features with shared-visual referents

Visual - Unshared Perspectives

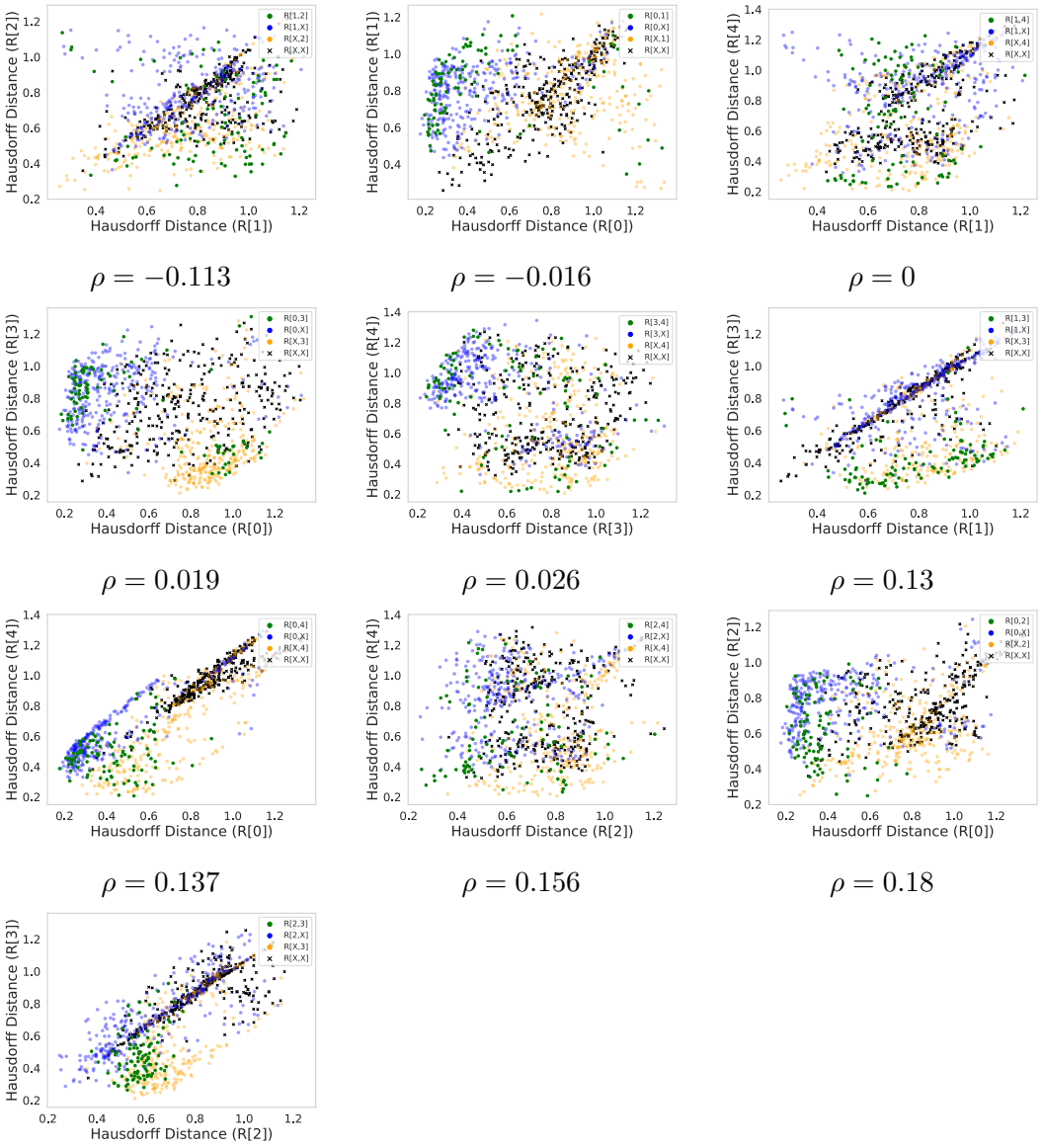


Figure A.12: Topographic maps and their associated topographic scores for each combination of features with unshared-visual referents

A.2.5 Composition Matrix Examples (Visual - Unshared Perspectives)

Figure A.13: Instances of descriptive utterances for referents from R_1 (blue frames) and R_2.

A.2.6 T-SNEs of embeddings (Visual - Unshared Perspectives)

R_2 referents & descriptive utterances

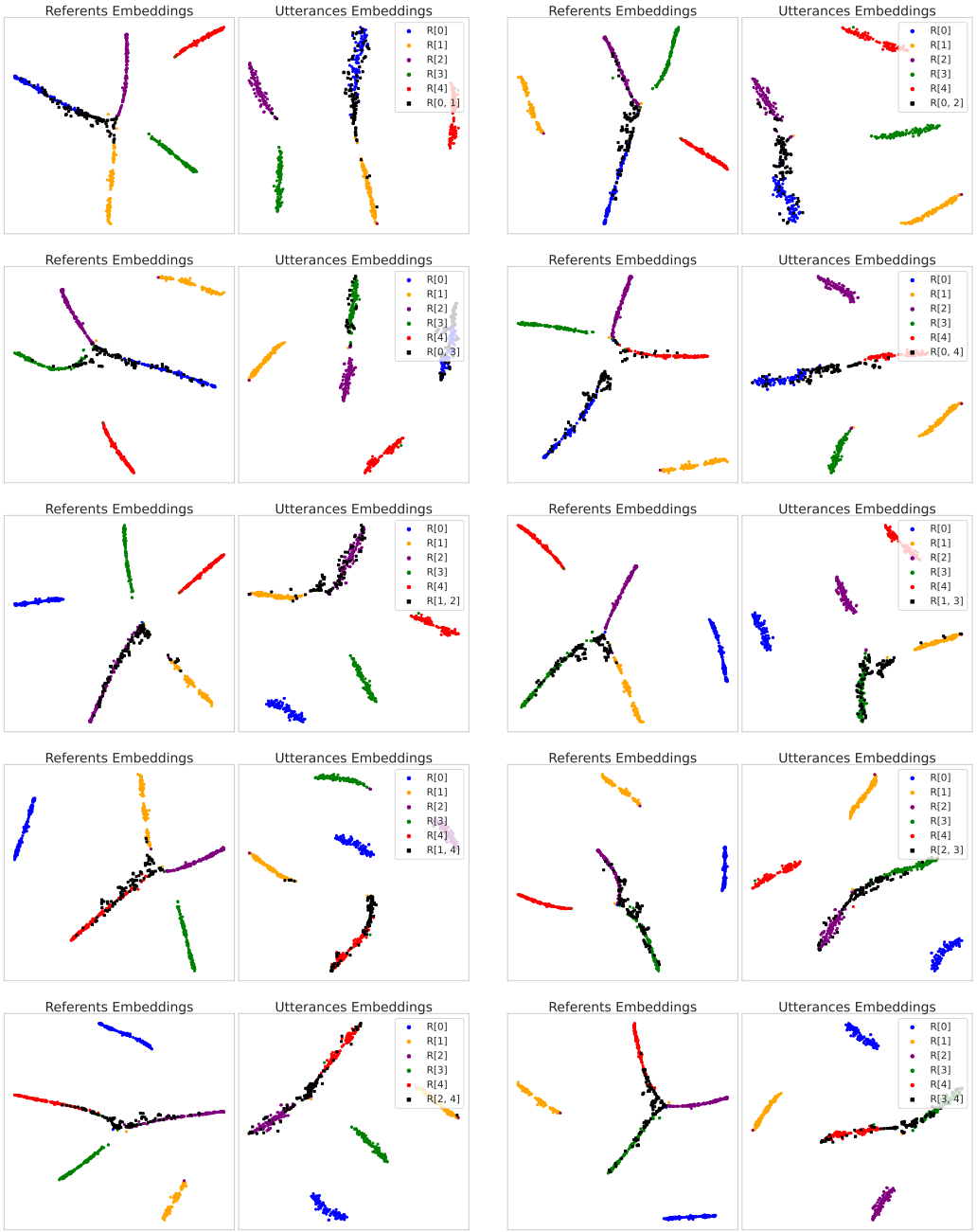


Figure A.14: **T-sne of referent and descriptive utterance embeddings.** Embeddings are computed for 100 perspectives of referents from R_2. Training conditions are unshared visual referents.

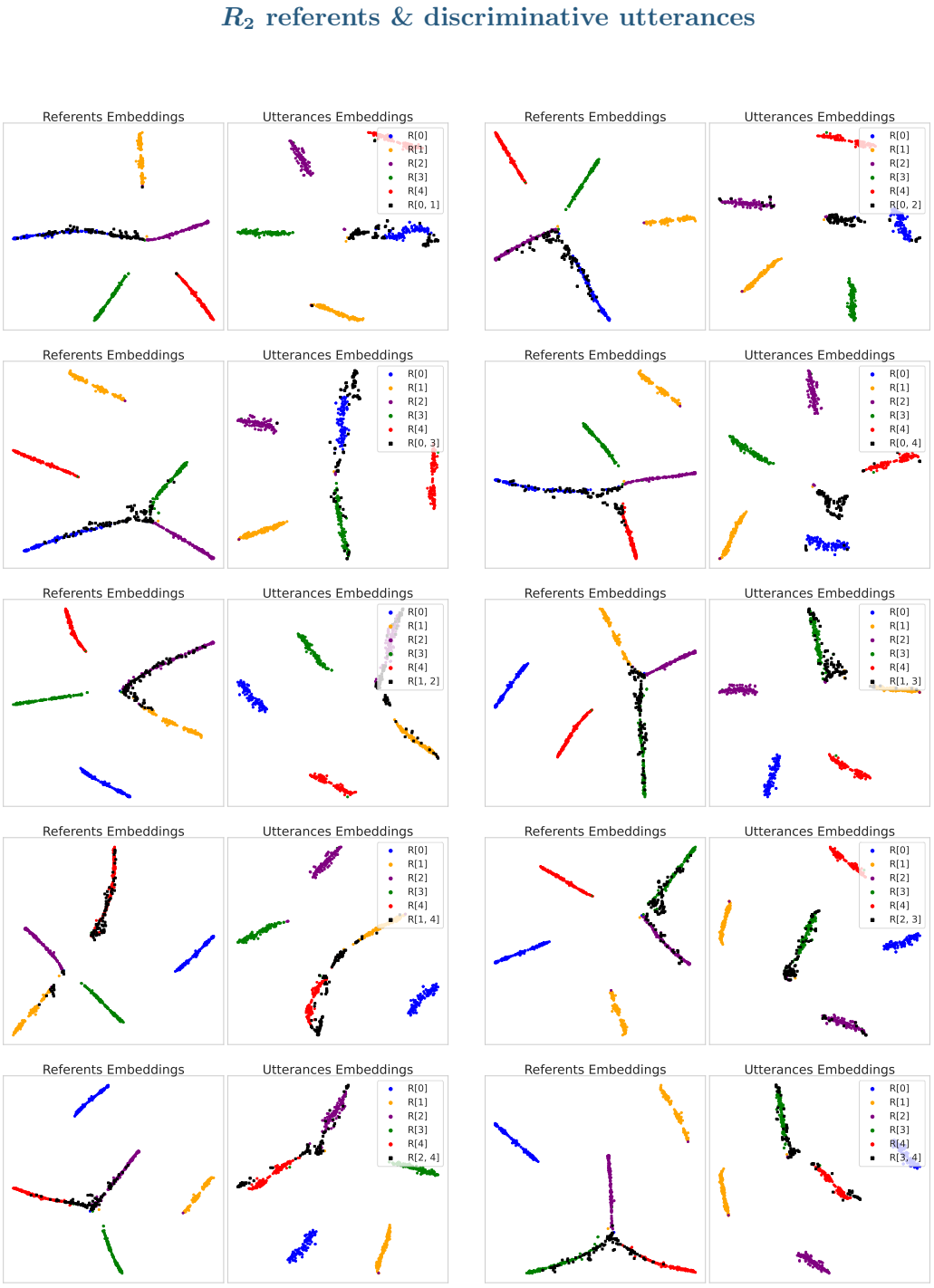


Figure A.15: **T-sne of referent and discriminative utterance embeddings.** Embeddings are computed for 100 perspectives of referents from R_2. Training conditions are unshared visual referents.

Appendix B

ABIG

This Supplementary Material provides additional derivations and implementation details. More specifically:

- Section B.1 provides additional diagrams illustrating the ABP problem and its position with respect to related settings.
- Section B.2 proposes the full derivation of the agents' MDP.
- Section B.3 provides additional details about ou algorithmic implementation.
- Section B.4 discusses the differences between ABP and Hierarchical/Feudal Reinforcement Learning.

B.1 Supplementary Sketches

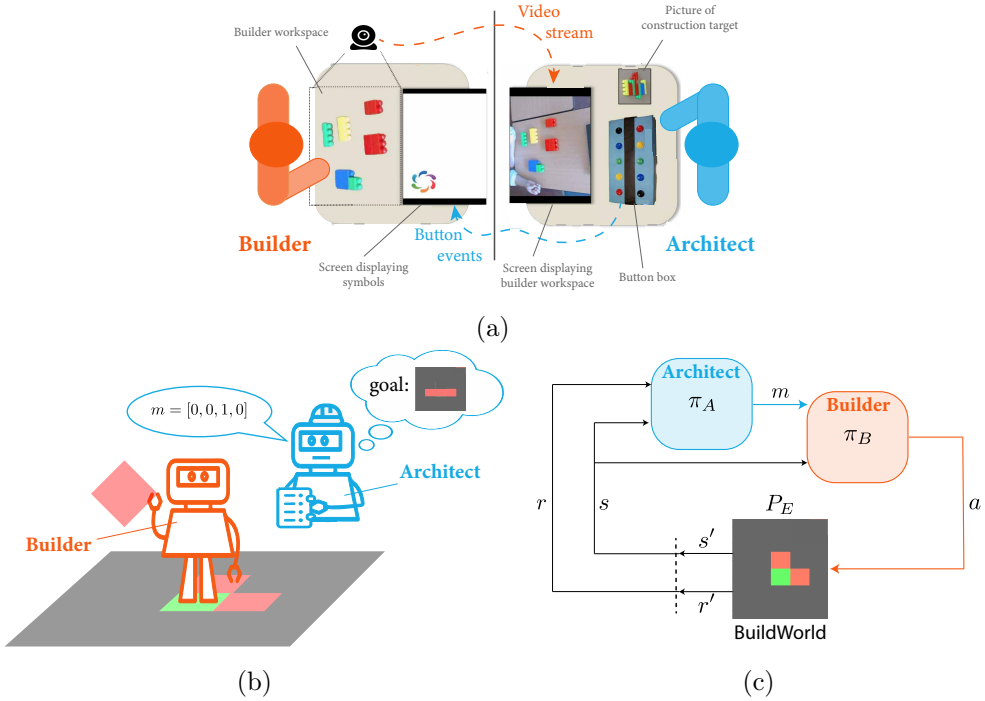
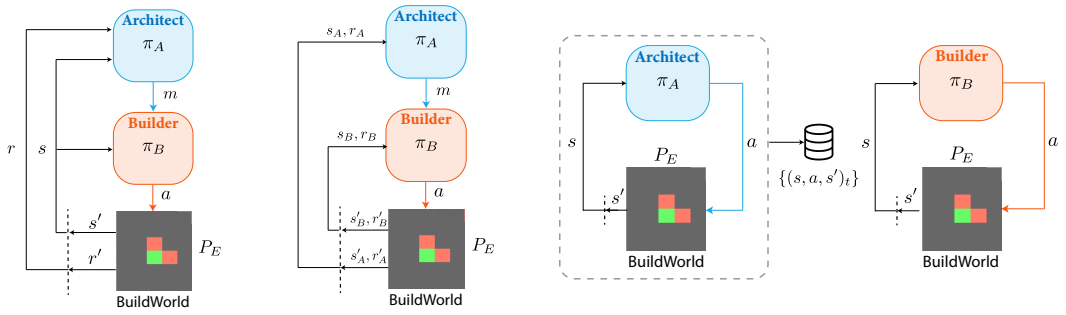


Figure B.1: (a) **Schematic view of the CoCo Game.** The architect and the builder should collaborate in order to build the construction target while located in different rooms. The architecture has a picture of the target while the builder has access to the blocks. The architect monitors the builder workspace via a camera (video stream) and can communicate with the builder only through the use of 10 symbols (button events). (b) **Schematic view of the Architect-Builder Problem.** The architect must learn how to use messages to guide the builder while the builder needs to learn to make sense of the messages in order to be guided by the architect. (c) **Interaction diagram between the agents and the environment in our proposed ABP.** The architect communicates messages (m) to the builder. Only the builder can act (a) in the environment. The builder conditions its action on the message sent by the builder ($\pi_B(a|s, m)$). The builder never perceives any reward from the environment



(a) (b) (c)

Figure B.2: (a) **Vertical view of the interaction diagram between the agents and the environment in our proposed ABP.** Only the architect perceives a reward signal r; (b) **Interaction diagram for a standard MARL modelization.** Both the architect and the builder have access to environmental rewards r_A and r_B. Which would contradict the fact that the builder ignores everything about the task at hand; (c) **Inverse Reinforcement Learning modelization of the ABP.** The architect needs to provide demonstrations. The architect does not exchange messages with the builder. The builder relies on the demonstrations $\{(s, a, s')_t\}$ to learn the desired behavior.

B.2 Analytical Description

Transition Probabilites from the architect point of view

Using the laws of total probabilities and conditional probabilities we have:

$$\begin{aligned} P_{_A}(s'|s,m) &= \sum_{a\in\mathcal{A}} P(s',a|s,m) \\ &= \sum_{a\in\mathcal{A}} P(s'|a,s,m)P(a|s,m) \\ &= \sum_{a\in\mathcal{A}} P_{_E}(s'|a,s)\tilde{\pi}_b(a|s,m) \end{aligned} \tag{B.1}$$

Where the final equality uses the knowledge that next-states only depends on states and builder's actions.

Reward function from the architect point of view

$$
\begin{aligned}
r_{A}(s,m,s') &\triangleq \mathbb{E}[R|s,m,s'] \\
&= \int_{\mathbb{R}} rP(r|s,m,s')dr \\
&= \int_{\mathbb{R}} r\sum_{a\in\mathcal{A}} P(r,a|s,m,s')dr \\
&= \int_{\mathbb{R}} r\sum_{a\in\mathcal{A}} P(r|s,m,a,s')P(a|s,m,s')dr \\
&= \int_{\mathbb{R}} r\sum_{a\in\mathcal{A}} P(r|s,a,s')\tilde{\pi}_b(a|s,m)dr \\
&= \sum_{a\in\mathcal{A}} \tilde{\pi}_b(a|s,m)\int_{\mathbb{R}} rP(r|s,a,s')dr \\
&= \sum_{a\in\mathcal{A}} \tilde{\pi}_b(a|s,m)r(s,a,s')
\end{aligned}
\tag{B.2}
$$

Transition function from the builder point of view

$$
\begin{aligned}
P(s',m'|s,m,a) &= P(m'|s',s,m,a)P(s'|s,m,a) \\
&= P(m'|s')P(s'|s,a) \\
&= \tilde{\pi}_{A}(m'|s')P_{E}(s'|s,a)
\end{aligned}
\tag{B.3}
$$

B.3 Practical Algorithm

Behavioral Cloning

The data-set is split into training (70%) and validation (30%) sets. If the validation accuracy does not improve during a *wait for* number of epochs the training is early stopped. For a training data-set $\mathcal{D} = \{(s,m,a)\}$ of size N the BC loss to minimize for a policy π_θ parametrized by θ is given by:

$$
J(\theta) = \frac{1}{N}\sum_{\mathcal{D}} -\log \pi_\theta(a|s,m) \tag{B.4}
$$

Monte-Carlo Tree Search

In the architect's MCTS, nodes are labeled by the environment's states and they are expanded by selecting messages. Selecting message m from a node with label s yields a builder action according to the architect's builder model $a \sim \tilde{\pi}_{B}(a|s,m)$, this sampled action in turn yields the label of the child node according to the environment's transition

model $s' \sim P_E(s'|s,a)$. We repeat this process until we select a message that was never selected from the current node or we sample a next state that does not correspond to a child node yet. In both of these cases, a new node has to be created. We estimate the value of the new node using an engineered heuristic that estimates the return of an optimal policy $\pi^*(a|s)$ from state s. This value is scaled down by a factor of 2 to avoid overestimation: the builder's policy may not allow the architect to have it follow π^*. This estimated value for a newly created node at depth l is back-propagated as a return to the parents node at depth k according to:

$$G^k = \sum_{\tau=0}^{l-1-k} \gamma^\tau r_{k+1+\tau} + \gamma^{l-k} v^l \qquad k = l, ..., 0 \tag{B.5}$$

where r_j is the reward collected from node at depth j to child node at depth $j+1$. From a node with label s we select messages according to the Upper Confidence Bound rule:

$$\begin{aligned} m &= \operatorname*{argmax}_m Q(s,m) + c\sqrt{\frac{\ln \sum_b N(s,b)}{N(s,m)}} \\ Q(s,m) &= \frac{\sum_i G_i(s,m)}{N(s,m)} \end{aligned} \tag{B.6}$$

where $N(s,m)$ is the number of times message m was selected from the node, $G_i(s,m)$ are the returns obtained from the node when selecting m and c is a constant set to $\sqrt{2}$. When the architect must choose a message from the environment state s, its policy $\pi_A(m|s)$ runs the above procedure from a root node labeled with the current environment state s. After expanding a budget b of nodes the architect picks the best message to send according to Eq. (B.6) applied to the root node. It is then possible to reuse the tree for the next action selection or to discard it, if a tree is reused its maximal depth should be constrained.

Hyper-parameters

sampling temperature	samples per iteration	learning rate	number of epochs	batch size
0.5	100	0.1	1000	50

Table B.1: Toy experiment hyper-parameters

budget	reuse tree	max tree depth
100	true	500

Table B.2: MCTS parameters

Sparse reward means that the architect receives 1 if the goal is achieved and 0 otherwise. Episodes per iterations are equally divided into the modelling and guiding frames.

episode len	grid size	reward	message
40	5×6 / (6 × 6)	sparse	one-hot

discount factor	episodes per iteration	vocab size	evaluation episode len
0.95	600	18 / (72)	40 / (60)

Table B.3: BuildWorld parameters for 3 blocks / (for 6 blocks if different)

learning rate	number of epochs	batch-size	wait for
5×10^{-4}	1000	256	300

Table B.4: Architect's BC parameters on BuildWorld for 3 blocks / (for 6 blocks if different)

learning rate	number of epochs	batch-size	wait for
1×10^{-4}	1000	256	300

Table B.5: Builder's BC parameters on BuildWorld for 3 blocks / (for 6 blocks if different)

Only the learning rates on BuildWorld were searched over with grid-searches. For BuildWorld with 3 blocks the searched range is $[5 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-5}]$ for both architect and builder (vocabulary size was fixed at 6). For 'grasp' with 6 blocks the searched range is $[1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}]$ for the architect and $[5 \times 10^{-4}, 1 \times 10^{-4}, 5 \times 10^{-5}]$ for the builder (vocabulary size was fixed at 72). The other hyper-parameters do not seem to have a major impact on the performance provided that:

- the MCTS hyper-parameters enable an agent that has access to the reward to solve the task.
- there is enough BC epochs to approach convergence.

Regarding the vocabulary size, the bigger the better (see experiments in Fig. 5.12).

Computing resources

A complete ABIG training can take up to 48 hours on a single modern CPU (`Intel E5-2683 v4 Broadwell @ 2.1GHz`). The presented results require approximately 700 CPU hours. For each training, the main computation cost comes from the MCTS planning during the guiding frames. The self-imitation and behavior modelling steps only account for a small fraction of the computation.

B.4 Related Work

In this section we develop the differences between ABP and Hierarchical/Feudal Reinforcement Learning more in detail.

Kulkarni et al. (2016) proposes to decompose a RL agent into a two-stage hierarchy with a meta-controller (or manager) setting the goals of a controller (or worker). The meta-controller is trained to select sequences of goals that maximize the environment reward while the controller is trained to maximize goal-conditioned intrinsic rewards. The definition of the goal-space as well as the corresponding hard-coded goal-conditioned reward functions are task-related design choices. In Vezhnevets et al. (2017), the authors propose a more general approach by defining goals as embeddings that directly modulate the worker's policy. Additionally, the authors define intrinsic rewards as the cosine distance between goals and embedded-state deltas (difference between the embedded-state at the moment the goal was given and the current embedded-state). Thus, goals can be interpreted as directions in embedding space. Nachum et al. (2018) build on a this idea but let go of the embedding transformation by considering goals as directions to reach and rewards as distances between state deltas and goals. These works tackle the single-agent learning problem and therefore allow the manager to directly influence the learning signal of the workers. However, in the multi-agent setting where agents are physically distinct, it is not possible for an agent to explicitly tweak another agent's learning algorithm. Instead, agents must communicate by influencing each other's observations instead of intrinsic rewards. Since it is designed to investigate the emergence of communication between agents, ABP lies in this latter multi-agent setting where agents can interact with one-another only through observations. This makes applying Feudal or Hierarchical methods to the ABP unfeasible as they are restricted to worker agents that directly receive rewards. In contrast, in ABP, the reward-less builder observes communication messages that, initially, have arbitrary meaning.

Appendix C

Grounding Spatio-Temporal Language with Transformers

C.1 Supplementary Methods

C.1.1 Input Encoding

We present the input processing in Fig. C.1. At each time step t, the body feature vector b_t and the object features vector $o_{i,t}$, $i = 1, 2, 3$ are encoded using two single-layer neural networks whose output are of size h. Similarly, each of the words of the sentence describing the trace (represented as one-hot vectors) is encoded and projected in the dimension of size h. We concatenate to the vector obtained a modality token m that defines if the output belongs to the scene $(1, 0)$ or to the description $(0, 1)$. We then feed the resulting vectors to a positional encoding that modulates the vectors according to the time step in the trace for b_t and $o_{i,t}$, $i = 1, 2, 3$ and according to the position of the word in the description for w_l.

We call the encoded body features $\hat{b}_t$ and it corresponds to $\hat{S}_{0,t}$ of the input tensor of our model (see Fig. 7.3 in the Main document). Similarly, $\hat{o}_{i,t}$, $i = 1, 2, 3$ are the encoded object features corresponding to $\hat{S}_{i,t}$, $i = 1, 2, 3$. Finally $\hat{w}_l$ are the encoded words and the components of tensor $\hat{W}$.

We call h the hidden size of our models and recall that $|\hat{b}_t| = |\hat{o}_{i,t}| = |\hat{w}_l| = h + 2$. This parameter is varied during the hyper-parameter search.

C.1.2 Details on Training Schedule

Implementation Details.

The architectures are trained via backpropagation using the Adam OptimizerKingma & Ba (2017). The data is fed to the model in batches of 512 examples for 150 000 steps. We use a modular buffer to sample an important variety of different descriptions in each batch and to impose a ratio of positive samples of 0.1 for each description in each batch.

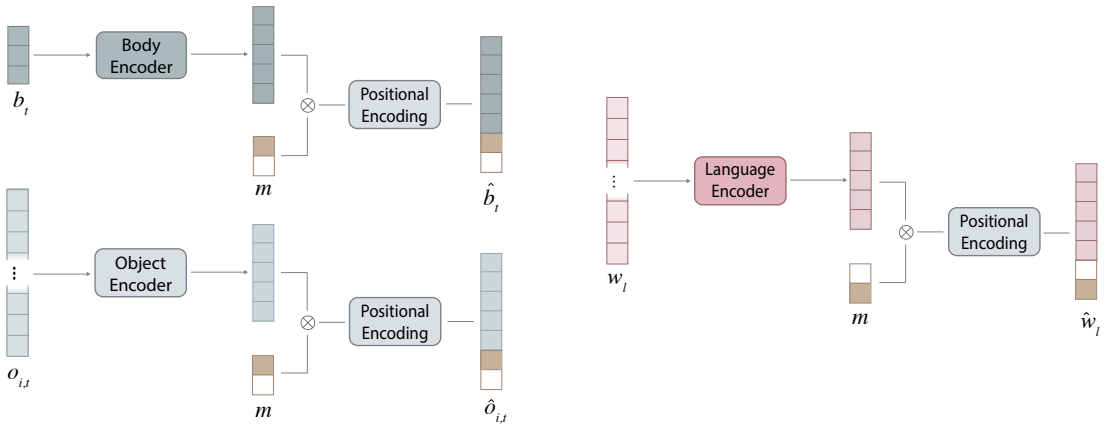


Figure C.1: **Diagram representing the projection of the inputs into the same dimension**

Model implementations.

We used the standard implementations of TransformerEncoderLayer and TransformerEncoder from pytorch version 1.7.1, as well as the default LSTM implementation. For initialization, we also use pytorch defaults.

Hyper-parameter search.

To pick the best set of parameters for each of our eight models, we train them on 18 conditions and select the best models. Note that each condition is run for 3 seeds and best models are selected according to their averaged F_1 score on randomly held-out descriptions (15% of the sentences in each category given in Table 7.1).

Best models.

Best models obtained thanks to the parameter search are given in Table C.1.

Model	**Learning rate**	**Model hyperparams**			
		hidden size	layer count	head count	param count
UT	1e-4	256	4	8	1.3M
UT-WA	1e-5	512	4	8	14.0M
TFT	1e-4	256	4	4	3.5M
TFT-WA	1e-5	512	4	8	20.3M
SFT	1e-4	256	4	4	3.5M
SFT-WA	1e-4	256	2	8	2.7M
LSTM-FLAT	1e-4	512	4	N/A	15.6M
LSTM-FACTORED	1e-4	512	4	N/A	17.6M

Table C.1: Hyperparameters for all models

Robustness to hyperparameters

For some models, we have observed a lack of robustness to hyperparameters during our search. This translated to models learning to predict all observation-sentence tuples as false since the dataset is imbalanced (the proportion of true samples is 0.1). This behavior was systematically observed with a series of models whose hyperparameters are listed in Table C.2. This happens with the biggest models with high learning rates, especially with the -WA variants.

Model	**Learning rate**	**Model hyperparams**		
		hidden size	layer count	head count
UT-WA	1e-4	512	4	4
UT-WA	1e-4	512	4	8
SFT	1e-4	512	4	4
SFT-WA	1e-4	512	4	8
SFT-WA	1e-4	512	2	4
SFT-WA	1e-4	512	4	4
TFT	1e-4	512	4	4
TFT-WA	1e-4	512	4	8
TFT-WA	1e-4	512	2	4
TFT-WA	1e-4	512	4	4

Table C.2: Models and hyperparameters collapsing into uniform false prediction.

C.2 Supplementary Results

C.2.1 Generalization to New Observations from Known Sentences

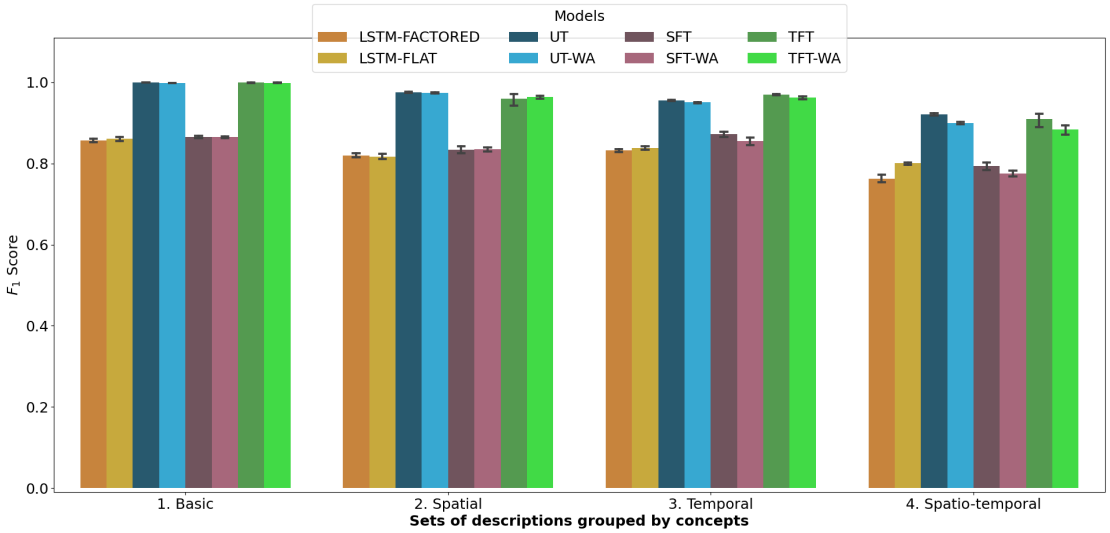


Figure C.2: F1 scores of all models on the train sentences with new observations.

In this section we shortly describe an additional evaluation setup we considered. We evaluate the model's f1-scores on sets of sentences that are seen as train sentences, but on newly generated observations. The results are plotted in Figure C.2.

C.2.2 Computing Resources

This work was performed using HPC resources from GENCI-IDRIS (Grant 2020-101594). We used 22k GPU-hours on nvidia-V100 GPUs for the development phase, hyperparameter search, and the main experiments.

Appendix D

IMAGINE

This supplementary material provides additional methods, results and discussion, as well as implementation details.

- Sec. D.1 gives an additional description of our setup and of the *Playground* environment.
- Sec. D.2 presents the set of testing goal description used for our generalization study.
- Sec. D.3 presents a focus on exploration and how it is influenced by goal imagination.
- Sec. D.4 presents a focus on the goal imagination mechanism we use for IMAGINE.
- Sec. D.5 presents a focus on the *Modular-Attention* architecture.
- Sec. D.6 presents a focus on the benefits of learning the reward function.
- Sec. D.7 provides additional visualization of the goal embeddings and the attention vectors.
- Sec. D.8 discusses the comparison with goal-as-state approaches.
- Sec. D.9 gives all necessary implementation details.

D.1 Additional Description of Playground and The Social Partner

Environment description

The environment is a 2D square: $[-1.2, 1.2]^2$. The agent is a disc of diameter 0.05 with an initial position $(0, 0)$. Objects have sizes uniformly sampled from $[0.2, 0.3]$ and their initial positions are randomized so that they are not in contact with each other. The agent has an action space of size 3 bounded in $[-1, 1]$. The first two actions control the agent's continuous 2D translation (bounded to 0.15 in any direction). The agent can grasp objects by getting in contact with them and closing its gripper (positive third action), unless it already has an object in hand. Objects include 10 animals, 10 plants, 10 pieces of furniture and 2 supplies. Admissible categories are *animal, plant, furniture, supply* and *living_thing* (animal or plant), see Fig. 7.2. Objects are assigned a color attribute (red, blue or green). Their precise color is a continuous RGB code uniformly

sampled from RGB subspaces associated with their attribute color. Each scene contains 3 of these procedurally-generated objects (see paragraph about the Social Partner below).

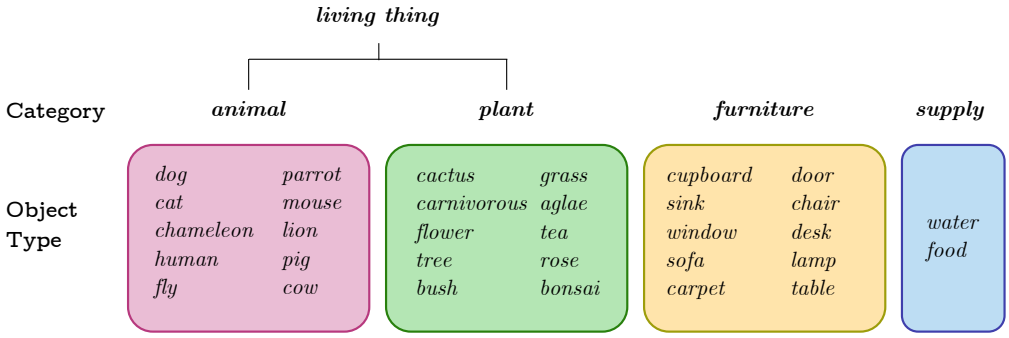


Figure D.1: Representation of possible objects types and categories.

Social Partner

SP has two roles:

- *Scene organization*: SP organize the scene according to the goal selected by the agent. When the agent selects a goal, it communicates it to SP. If the goal starts by the word *grow*, SP adds a procedurally-generated supply (water or food for animals, water for plants) of any size and color to the scene. If the goal contains an object (e.g. *red cat*), SP adds a corresponding object to the scene (with a procedurally generated size and RGB color). Remaining objects are generated procedurally. As a result, the objects required to fulfill a goal are always present and the scene contains between 1 (*grow* goals) and 3 (*go* goals) random objects. Note that all objects are procedurally generated (random initial position, RGB color and size).
- *Scene description*: SP provides NL descriptions of interesting outcomes experienced by the agent at the end of episodes. It takes the final state of an episode ($\mathbf{s}_T$) as input and returns matching NL descriptions: $\mathcal{D}_{\text{SP}}(\mathbf{s}_T) \subset \mathcal{D}^{\text{SP}}$. When SP provides *descriptions*, the agent considers them as targetable *goals*. This mapping $\mathcal{D}^{\text{SP}} \rightarrow \mathcal{G}^{\text{train}}$ simply consists in removing the first *you* token (e.g. turning *you grasp red door* into the goal *grasp red door*). Given the set of previously discovered goals ($\mathcal{G}_{\text{known}}$) and new descriptions $\mathcal{D}_{\text{SP}}(\mathbf{s}_T)$, the agent infers the set of goals that were not achieved: $\mathcal{G}_{\text{na}}(\mathbf{s}_T) = \mathcal{G}_{\text{known}} \setminus \mathcal{D}_{\text{SP}}(\mathbf{s}_T)$, where $\setminus$ indicates the complement.

D.2 Testing Set of Goals for Generalization

Because scenes are procedurally-generated, $\overline{\texttt{SR}}$ computed on $\mathcal{G}^{\text{train}}$ measures the generalization to new states. When computed on $\mathcal{G}^{\text{test}}$, however, $\overline{\texttt{SR}}$ measures both this state generalization and the generalization to new goal descriptions from $\mathcal{G}^{\text{test}}$. As $\overline{\texttt{SR}}_{\text{train}}$ is almost perfect, this section focuses solely on generalization in the language space: $\overline{\texttt{SR}}_{\text{test}}$.

Table D.1: Testing goals in $\mathcal{G}^{\text{test}}$, by type.

Type 1	*Grasp blue door*, *Grasp green dog*,*Grasp red tree*, *Grow green dog*
Type 2	*Grasp any flower*, *Grasp blue flower*, *Grasp green flower*, *Grasp red flower*, *Grow any flower*, *Grow blue flower*, *Grow green flower*, *Grow red flower*
Type 3	*Grasp any animal*, *Grasp blue animal*, *Grasp green animal*, *Grasp red animal*
Type 4	*Grasp any fly*, *Grasp blue fly*, *Grasp green fly*, *Grasp red fly*
Type 5	*Grow any algae*, *Grow any bonsai*, *Grow any bush*, *Grow any cactus* *Grow any carnivorous*, *Grow any grass*, *Grow any living_thing*, *Grow any plant* *Grow any rose*, *Grow any tea*, *Grow any tree*, *Grow blue algae* *Grow blue bonsai*, *Grow blue bush*,*Grow blue cactus*, *Grow blue carnivorous* *Grow blue grass*, *Grow blue living_thing*, *Grow blue plant*, *Grow blue rose* *Grow blue tea*, *Grow blue tree*,*Grow green algae*, *Grow green bonsai* *Grow green bush*, *Grow green cactus*, *Grow green carnivorous*, *Grow green grass* *Grow green living_thing*, *Grow green plant*, *Grow green rose*, *Grow green tea* *Grow green tree*, *Grow red algae*, *Grow red bonsai*, *Grow red bush* *Grow red cactus*, *Grow red carnivorous*, *Grow red grass*, *Grow red living_thing* *Grow red plant*, *Grow red rose*, *Grow red tea*, *Grow red tree*

D.3 Focus on Exploration

Interesting Interactions

Interesting interactions are trajectories of the agent that humans could infer as goal-directed. If an agent brings water to a plant and grows it, it makes sense for a human. If it then tries to do this for a lamp, it also feels goal-directed, even though it does not work. This type of behavior characterizes the penchant of agents to interact with objects around them, to try new things and, as a result, is a good measure of exploration.

Sets of interesting interactions

We consider three sets of interactions: 1) interactions related to training goals; 2) to testing goals; 3) the extra set. This *extra set* contains interactions where the agent brings water or food to a piece of furniture or to another supply. Although such behaviors do not achieve any of the goals, we consider them as interesting exploratory behaviors. Indeed, they testify that agents try to achieve imagined goals that are meaningful from the point of view of an agent that does not already know that doors cannot be grown, i.e. corresponding to a meaningful form of generalization after discovering that animals or plants can be grown (e.g. *grow any door*).

The Interesting Interaction Count metric

We count the number of interesting interactions computed over all final transitions from the last 600 episodes (1 epoch). Agents do not need to target these interactions, we just report the number of times they are experienced. Indeed, the agent does not have to target a particular interaction for the trajectory to be interesting from an exploratory point of view. The HER mechanism ensures that these trajectories can be replayed to learn about any goal, imagined or not. Computed on the extra set, the *Interesting Interaction Count* (I2C) is the number of times the agent was found to bring supplies to a furniture or to other supplies over the last epoch:

$$\text{I2C}_{\text{extra}} = \sum_{i \in \mathcal{I}=\mathcal{G}_{\text{extra}}} \sum_{t=1}^{600} \delta_{i,t},$$

where $\delta_{i,t} = 1$ if interaction i was achieved in episode t, 0 otherwise and $\mathcal{I}$ is the set of interesting interactions (here from the extra set) performed during an epoch.

Agents that are allowed to imagine goals achieve higher scores in the testing and extra sets of interactions, while maintaining similar exploration scores on the training set, see Figures D.2a to D.2c.

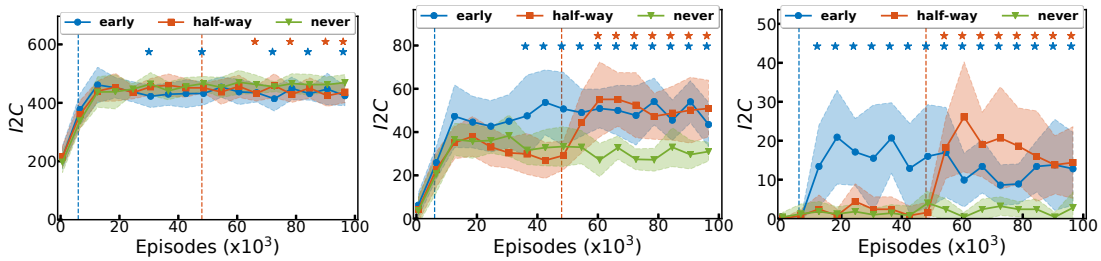


Figure D.2: **Exploration metrics** (a) Interesting interaction count (I2C) on training set, (b) I2C on testing set, (c) I2C on extra set. Goal imagination starts early (vertical blue line), half-way (vertical orange line) or does not start (*no imagination* baseline in green).

D.4 Focus on Goal Imagination

Algorithm 8 presents the algorithm underlying our goal imagination mechanism. This mechanism is inspired from the *Construction Grammar* (CG) literature and generates new sentences by composing known ones (Goldberg, 2003). It computes sets of equivalent words by searching for sentences with an edit distance of 1: sentences where only one word differs. These words are then labelled equivalent, and can be substituted in known sentences. Note that the goal imagination process filters goals that are already known. Although all sentences from $\mathcal{G}^{\text{train}}$ can be imagined, there are filtered out of the imagined goals as they are discovered. Imagining goals from $\mathcal{G}^{\text{train}}$ before they are discovered drives the exploration of IMAGINE agents. In our setup, however, this effect remains marginal as all the goals from $\mathcal{G}^{\text{train}}$ are discovered in the first epochs (see Fig. D.4).

Algorithm 8: Goal Imagination. The edit distance between two sentences refers to the number of words to modify to transform one sentence into the other.

Input: $\mathcal{G}_{\text{known}}$ (discovered goals)
Initialize: *word_eq* (list of sets of equivalent words, empty)
Initialize: *goal_template* (list of template sentences used for imagining goals, empty)
Initialize: $\mathcal{G}_{\text{im}}$ (empty)
for g_{NL} in $\mathcal{G}_{\text{known}}$ **do** {Computing word equivalences}
  new_goal_template = True
  for g_m in *goal_template* **do**
    if edit_distance(g_{NL}, g_m) < 2 **then**
      new_goal_template = False
      if edit_distance(g_{NL}, g_m) == 1 **then**
        $w_1, w_2 \leftarrow$ *get_non_matching_words*(g_{NL}, g_m)
        if w_1 and w_2 not in any of *word_eq* sets **then**
          word_eq.add($\{w_1, w_2\}$)
        else
          for *eq_set* in *word_eq* **do**
            if $w_1 \in$ *eq_set* or $w_2 \in$ *eq_set* **then**
              eq_set = *eq_set* $\cup \{w_1, w_2\}$
            end if
          end for
        end if
      end if
    end if
  end for
  if *new_goal_template* **then**
    goal_template.add(g_{NL})
  end if
end for
for *g* in *goal_template* **do** {Generating new sentences}
  for *w* in *g* **do**
    for *eq_set* in *word_eq* **do**
      if *w* ∈ *eq_set* **then**
        for *w′* in *eq_set* **do**
          $g_{im} \leftarrow$ replace(*g*, *w*, *w′*)
          if $g_{im} \notin \mathcal{G}_{\text{known}}$ **then**
            $\mathcal{G}_{\text{im}} = \mathcal{G}_{\text{im}} \cup \{g_{im}\}$
          end if
        end for
      end if
    end for
  end for
end for
$\mathcal{G}_{\text{im}} = \mathcal{G}_{\text{im}} \setminus \mathcal{G}_{\text{known}}$ {filtering known goals.}

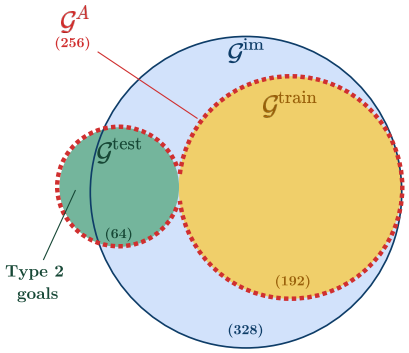


Figure D.3: Venn diagram of goal spaces.

Table D.2: All imaginable goals $\mathcal{G}^{\text{im}}$ generated by the Construction Grammar Heuristic.

Goals from $\mathcal{G}^{\text{train}}$	$\mathcal{G}^{\text{train}}$. (Note that known goals are filtered from the set of imagined goals. However, any goal from $\mathcal{G}^{\text{train}}$ can be imagined before it is encountered in the interaction with SP.)
Goals from $\mathcal{G}^{\text{test}}$	All goals from Type 1, 3, 4 and 5, see Table D.1
Syntactically incorrect goals	*Go bottom top*, *Go left right*, *Grasp red blue thing*, *Grow blue red thing*, *Go right left*, *Go top bottom*, *Grasp green blue thing*, *Grow green red thing*, *Grasp green red thing* *Grasp blue green thing*, *Grasp blue red thing*, *Grasp red green thing*.
Syntactically correct but unachievable goals	*Go center bottom*, *Go center top*, *Go right center*, *Go right bottom*, *Go right top*, *Go left center*, *Go left bottom*, *Go left top*, *Grow green cupboard*, *Grow green sink*, *Grow blue lamp*, *Go center right*, *Grow green window*, *Grow blue carpet*, *Grow red supply*, *Grow any sofa*, *Grow red sink*, *Grow any chair*, *Go top center*, *Grow blue table*, *Grow any door*, *Grow any lamp*, *Grow blue sink*, *Go bottom center*, *Grow blue door*, *Grow blue supply*, *Grow green carpet*, *Grow blue furniture*, *Grow green supply*, *Grow any window*, *Grow any carpet*, *Grow green furniture*, *Grow green chair*, *Grow green food*, *Grow any cupboard*, *Grow red food*, *Grow any table*, *Grow red lamp* , *Grow red door*, *Grow any food*, *Grow blue window*, *Grow green sofa*, *Grow blue sofa*, *Grow blue desk*, *Grow any sink*, *Grow red cupboard*, *Grow green door*, *Grow red furniture*, *Grow blue food*, *Grow red desk* , *Grow red table*, *Grow blue chair*, *Grow red sofa*, *Grow any furniture*, *Grow red window*, *Grow any desk*, *Grow blue cupboard*, *Grow red chair*, *Grow green desk*, *Grow green table*, *Grow red carpet*, *Go center left*, *Grow any supply*, *Grow green lamp*, *Grow blue water*, *Grow red water*, *Grow any water*, Grow green water, *Grow any water*, Grow green water.

Imagined goals

We run our goal imagination mechanism based on the Construction Grammar Heuristic (CGH) from $\mathcal{G}^{\text{train}}$. After filtering goals from $\mathcal{G}^{\text{train}}$, this produces 136 new imagined sentences. Table D.2 presents the list of these goals while Fig. D.3 presents a Venn diagram of the various goal sets. Among these 136 goals, 56 belong to the testing set $\mathcal{G}^{\text{test}}$. This results in a coverage of 87.5% of $\mathcal{G}^{\text{test}}$, and a precision of 45%. In goals that do not belong to $\mathcal{G}^{\text{test}}$, goals of the form *Grow* + {*any*} ∪ **color** + **furniture** ∪ **supplies** (e.g. *Grow any lamp*) are *meaningful* to humans, but are not achievable in the environment (*impossible*).

Variants of goal imagination mechanisms

Main Sec. 8.5.3 investigates variants of our goal imagination mechanisms:

1. *Lower coverage:* To reduce the coverage of CGH while maintaining the same precision, we simply filter half of the goals that would have been imagined by CGH. This filtering is probabilistic, resulting in different imagined sets for different runs. It happens online, meaning that the coverage is always half of the coverage that CGH would have had at the same time of training.

2. *Lower precision:* To reduce precision while maintaining the same coverage, we sample a random sentence (random words from the words of $\mathcal{G}^{\text{train}}$) for each goal imagined by CGH that does not belong to $\mathcal{G}^{\text{test}}$. Goals from $\mathcal{G}^{\text{test}}$ are still imagined via the CGH mechanism. This variants only doubles the imagination of sentences that do not belong to $\mathcal{G}^{\text{test}}$.

3. *Oracle:* Perfect precision and coverage is achieved by filtering the output of CGH, keeping only goals from $\mathcal{G}^{\text{test}}$. Once the 56 goals that CGH can imagine are imagined, the oracle variants adds the 8 remaining goals: those including the word *flower* (Type 2 generalization).

4. *Random goals:* Each time CGH would have imagined a new goal, it is replaced by a randomly generated sentence, using words from the words of $\mathcal{G}^{\text{train}}$.

Note that all variants imagine goals at the same speed as the CGH algorithm. They simply filter or add noise to its output, see Fig. D.4.

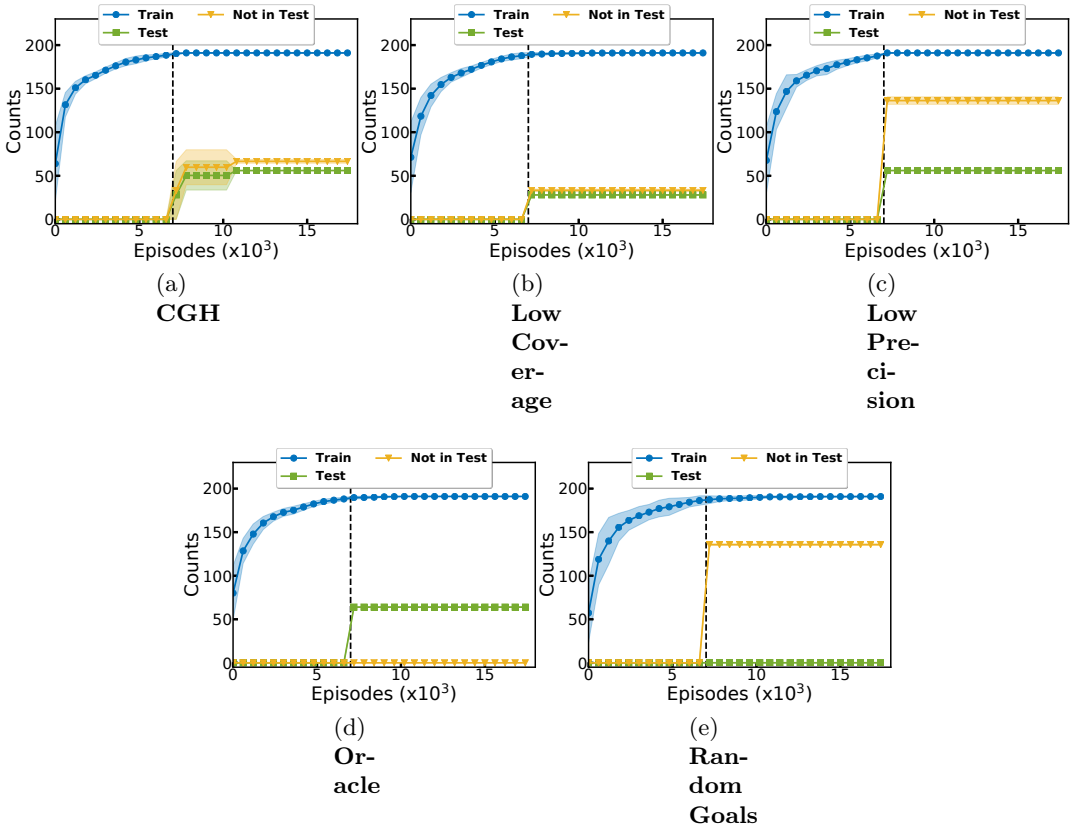


Figure D.4: **Evolution of known goals for various goal imagination mechanisms.** All graphs show the evolution of the number of goals from $\mathcal{G}^{\text{train}}$, $\mathcal{G}^{\text{test}}$ and others in the list of known goals $\mathcal{G}_{\text{known}}$. We zoom on the first epochs, as most goals are discovered and invented early. Vertical dashed line indicates the onset of goal imagination. (a) CGH; (b) Low Coverage; (c) Low precision; (d) Oracle; (e) Random Goals.

Effect of low coverage on generalization

In Main Sec. 8.5.3, we compare our goal imagination mechanism to a *Low Coverage* variant that only covers half of the proportion of $\mathcal{G}^{\text{test}}$ covered by CGH (44%). Fig. D.5 shows that the generalization performance on goals from $\mathcal{G}^{\text{test}}$ that the agent imagined (n-shot generalization, blue) are not significantly higher than the generalization performance on goals from $\mathcal{G}^{\text{test}}$ that were not imagined (zero-shot generalization). As they are both significantly higher than the *no imagination* baseline, this implies that training on imagined goals boosts zero-shot generalization on similar goals that were not imagined.

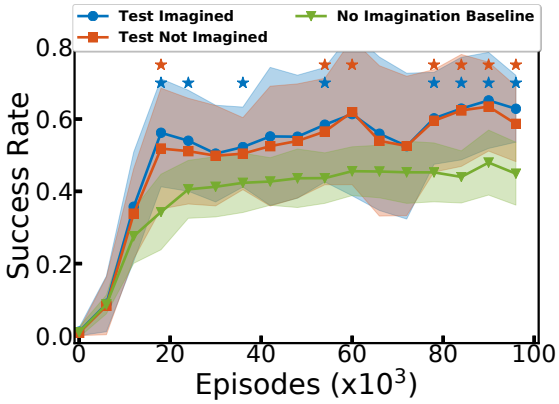
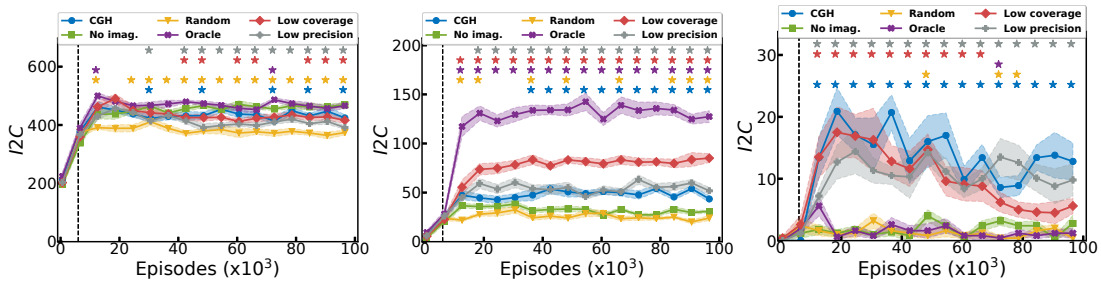


Figure D.5: **Zero-shot versus n-shot.** We look at the *Low Coverage* variant of our goal imagination mechanism that only covers 43.7% the test set with a 45% precision. We report success rates on testing goals of Type 5 (*grow + plant*) and compare with the *no imagination* baseline (green). We split in two: goals that were imagined (blue), and goals that were not (orange).

Details on the impacts of various goal imagination mechanisms on exploration

Fig. D.6 presents the I2C exploration scores on the training, testing and extra sets for the different goal imagination mechanisms introduced in Main Sec. 8.5.3. Let us discuss each of these scores:

1. *Training interactions.* In Fig. D.6a, we see that decreasing the precision (Low Precision and Random Goal conditions) affects exploration on interactions from the training set, where it falls below the exploration of the *no imagination* baseline. This is due to the addition of meaningless goals forcing agent to allow less time to meaningful interactions relatively.
2. *Testing interactions.* In Fig. D.6b, we see that the highest exploration scores on interactions from the test set comes from the oracle. Because it shows high coverage and precision, its spends more time on the diversity of interactions from the testing set. What is more surprising is the exploration score of the low coverage condition, higher than the exploration score of CGH. With an equal precision, CGH should show better exploration, as it covers more test goals. However, the *Low Coverage* condition, by spending more time exploring each of its imagined goals (it imagined fewer), probably learned to master them better, increasing the robustness of its behavior towards those. This insight advocates for the use of goal selection methods based on learning progress (Forestier & Oudeyer, 2016; Colas et al., 2019a). Agents could estimate their learning progress on imagined goals using their internal reward function and its zero-shot generalization. Focusing on goals associated to high learning progress might help agents filter goals they can learn about from others.
3. *Extra interactions.* Fig. D.6c shows that only the goal imagination mechanisms that invent goals not covered by the testing set manage to boost exploration in this extra set. The oracle perfectly covers the testing set, but does not generate goals related to other objects (e.g. *grow any lamp*).



(a) **I2C on $\mathcal{G}^{\text{train}}$** (b) **I2C on $\mathcal{G}^{\text{test}}$** (c) **I2C on $\mathcal{G}^{\text{extra}}$**

Figure D.6: **Exploration metrics for different goal imagination mechanisms**: (a) Interesting interaction count (I2C) on training set, I2C on testing set, (c) I2C on extra set. Goal imagination starts early (vertical line), except for the *no imagination* baseline (green). Standard errors of the mean plotted for clarity (as usual, 10 seeds).

D.5 Focus on Architectures

This section compares our proposed object-based modular architecture MA for the policy and reward function to a flat architecture that does not use inductive biases for efficient skill transfer. We hypothesize that only the object-based modular architectures enable a generalization performance that is sufficient for the goal imagination to have an impact on generalization and exploration. Indeed, when generalization abilities are low, agents cannot evaluate their performance on imagined goals and thus, cannot improve.

Preliminary study of the reward function architecture

We first compared the use of modular and flat architectures for the reward function ($\text{MA}^{\mathcal{R}}$ vs $\text{FA}^{\mathcal{R}}$ in Fig. D.7). This experiment was conducted independently from policy learning, in a supervised setting. We use a dataset of 50×10^3 trajectories and associated goal descriptions collected using a pre-trained policy. To closely match the training conditions of IMAGINE, we train the reward function on the final states s_T and test it on any states s_t, $t = [1, .., T]$ of other episodes. Table D.3 provides the F_1 score computed at convergence on $\mathcal{G}^{\text{train}}$ and $\mathcal{G}^{\text{test}}$ for the two architectures.

Table D.3: Reward function architectures performance.

	$F_{1_{\text{train}}}$	$F_{1_{\text{test}}}$
$\text{MA}^{\mathcal{R}}$	0.98 ± 0.02	0.64 ± 0.22
$\text{FA}^{\mathcal{R}}$	0.60 ± 0.10	0.22 ± 0.05

$\text{MA}^{\mathcal{R}}$ outperforms $\text{FA}^{\mathcal{R}}$ on both the training and testing sets. In addition to its poor generalization performance, $\text{FA}^{\mathcal{R}}$'s performance on the training set are too low to support policy learning. As a result, the remaining experiments in this paper use the $\text{MA}^{\mathcal{R}}$ architecture for all reward functions. Thereafter, MA is always used for the reward function and the terms MA and FA refer to the architecture of the policy.

Architectures representations

The combination of MA for the reward function and either MA or FA for the policy are represented in Fig. **??**.

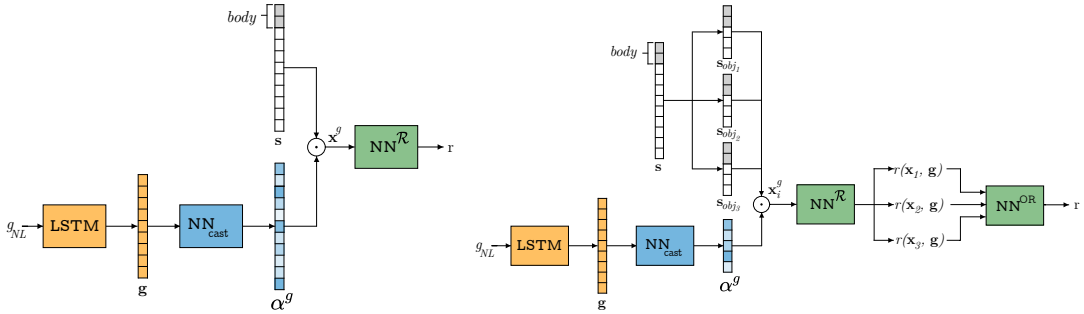
Policy architecture comparison

Table D.4 shows that MA significantly outperforms FA on both the training and testing sets at convergence. Fig. D.8a clearly shows an important gap between the generalization performance of the modular and the flat architecture. In average, less than 20% of the testing goals can be achieved with FA when MA masters half of them without imagination. Moreover, there is no significant difference between the never and the early imagination conditions for the flat architecture. The generalization boost enabled by the imagination

is only observable for the modular architecture (see Main Table 8.2). Fig. D.8c and D.8d support similar conclusions for exploration: only the modular architecture enable goal imagination to drive an exploration boost on the testing and extra sets of interactions.

Table D.4: Architectures performance. Both p-values $< 10^{-10}$.

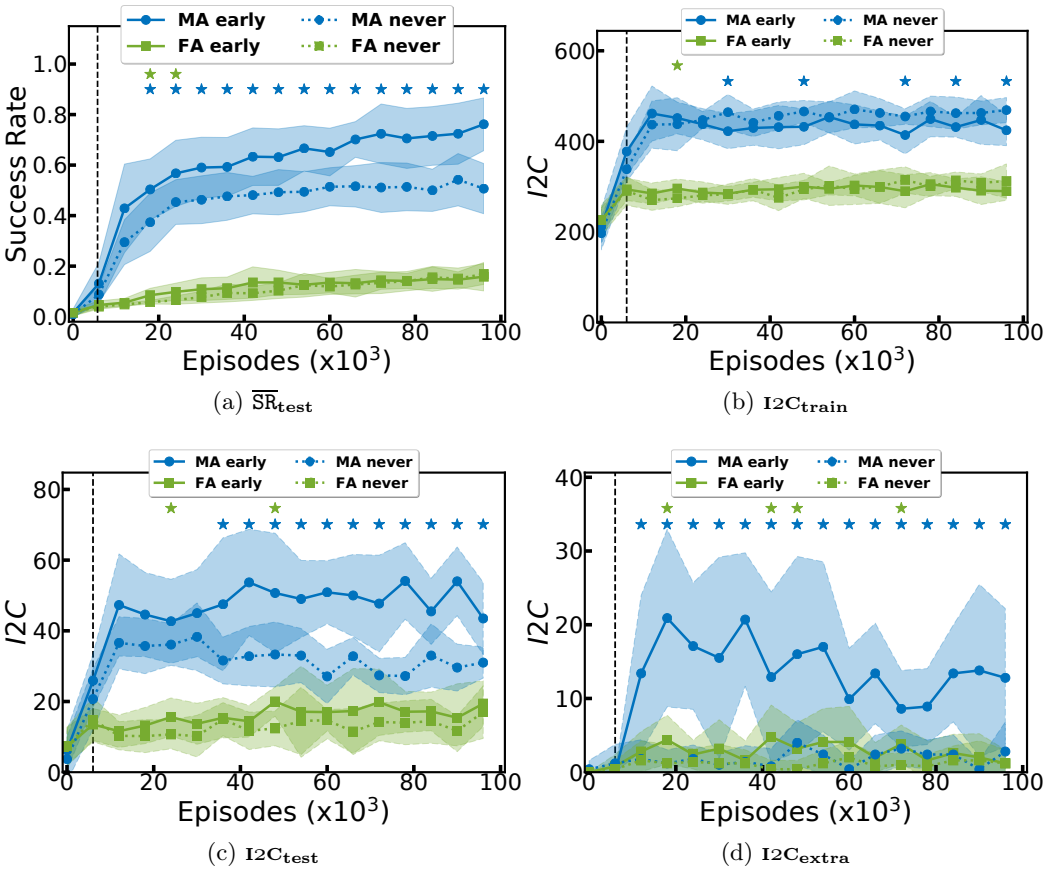
	$\overline{\text{SR}}_{\text{train}}$	$\overline{\text{SR}}_{\text{test}}$
MA	0.95 ± 0.05	0.76 ± 0.10
FA	0.40 ± 0.13	0.16 ± 0.06



(a) FA$^{\mathcal{R}}$ (b) MA$^{\mathcal{R}}$

Figure D.7: **Reward function architectures**: (a) *Flat-attention* reward function (FA$^{\mathcal{R}}$) and (b) *Modular-attention* reward function (MA$^{\mathcal{R}}$). We use MA$^{\mathcal{R}}$ for all experiments except for the experiment in Table D.3

In preliminary experiments, we tested a *Flat-Concatenation* (FC) architecture where the gated attention mechanism was replaced by a simple concatenation of goal encoding to the state vector. We did not found signficant difference with respect to FA. We chose to pursue with the attention mechanism, as it improves model interpretability (see Additional Visualization D.7).



(a) $\overline{\text{SR}}_{\text{test}}$ (b) $\text{I2C}_{\text{train}}$

(c) I2C_{test} (d) $\text{I2C}_{\text{extra}}$

Figure D.8: **Policy architecture comparison:** (a) $\overline{\text{SR}}$ on $\mathcal{G}^{\text{test}}$ for the FA and MA architectures when the agent starts imagining goals early (plain, after the black vertical dashed line) or never (dashed). (b, c, d) I2C on interactions from the training, testing and extra sets respectively. Imagination is performed using CGH. Stars indicate significant differences between CGH and the corresponding *no imagination* baseline.

D.6 Focus on Reward Function

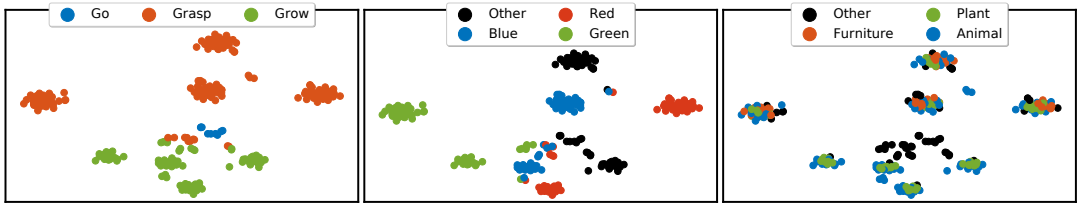
Our IMAGINE agent is autonomous and, as such, needs to learn its own reward function. It does so by leveraging a weak supervision from a social partner that provides descriptions in a simplified language. This reward function can be used for many purposes in the architecture. This paper leverages some of these ideas (the first two), while others are left for future work (the last two):

- **Behavior Adaptation.** As Main Sec. 8.5.1 showed, the reward function enables agents to adapt their behavior with respect to imagined goals. Whereas the zero-shot generalization pushed agents to grow plants with food and water with equal probability, the reward function helped agents to correct that behavior towards more water.
- **Guiding Hindsight Experience Replay (HER).** In multi-goal RL with discrete sets of goals, HER is traditionally used to modify transitions sampled from the replay buffer. It replaces originally targeted goals by others randomly selected from the set of goals (Andrychowicz et al., 2017a; **?**). This enables to transfer knowledge between goals, reinterpreting trajectories in the light of new goals. In that case, a reward function is required to compute the reward associated to that new transition (new goal). To improve on random goal replay, we favor goal substitution towards goals that actually match the state and have higher chance of leading to rewards. In IMAGINE, we scan a set of 40 goal candidates for each transition, and select substitute goals that match the scene when possible, with probability $p = 0.5$.
- **Exploring like Go-Explore.** In Go-Explore (**?**), agents first reach a goal state, then start exploring from there. We could reproduce that behavior in our IMAGINE agents with our internal reward function. The reward function would scan each state during the trajectory. When the targeted goal is found to be reached, the agent could switch to another goal, add noise on its goal embedding, or increase the exploration noise on actions. This might enable agents to explore sequences of goal-directed behaviors. We leave the study of this mechanism for future work.
- **Filtering of Imagined Goals.** When generating imagined goals, agents also generate meaningless goals. Ideally, we would like agents to filter these from meaningful goals. Meaningful goals, are goals the agent can interpret with its reward function, goals from which it can learn directed behavior. They are interpreted from known related goals via the generalization of the reward function. If we consider an ensemble of reward functions, chances are that all reward functions in the ensemble will agree on the interpretation of meaningful imagined goals. On the other hand, they might disagree on meaningless goals, as their meanings might not be as easily derived from known related goals. Using an ensemble of reward function may thus help agents filter meaningful goals from meaningless ones. This could be done by labeling a dataset of trajectories with positive or negative rewards and comparing results between reward functions, effectively computing agreement measures for each imagined goals. Having an efficient filtering mechanism would drastically improve the efficiency of goal imagination, as Main Sec. 8.5.3 showed that the ratio of meaningful goals determines generalizations performance. This is also left for future work.

D.7 Additional Visualizations

Visualizing Goal Embedding

To analyze the goal embeddings learned by the language encoder L_e, we perform a t-SNE using 2 components, perplexity 20, a learning rate of 10 for 5000 iterations. Fig. D.9 presents the resulting projection for a particular run. The embedding seems to be organized mainly in terms of motor predicates (D.9a), then in terms of colors (D.9b). Object types or categories do not seem to be strongly represented (D.9c).



(a) (b) (c)

Figure D.9: **t-SNE of Goal Embedding.** The same t-SNE is presented, with different color codes (a) predicates, (b) colors, (c) object categories.

Visualizing Attention Vectors

In the *modular-attention* architectures for the reward function and policy, we train attention vectors to be combined with object-specific features using a gated attention mechanism. In each architecture, the attention vector is shared across objects (permutation invariance). Fig. D.10 presents examples of attention vectors for the reward function (D.10a) and for the policy (D.10b) at the end of training. These attention vectors highlight relevant parts of the object-specific sub-state depending on the NL goal:

- When the sentence refers to a particular object type (e.g. *dog*) or category (e.g. *living thing*), the attention vector suppresses the corresponding object type(s) and highlights the complement set of object types. If the object does not match the object type or category described in the sentence, the output of the Hadamard product between object types and attention will be close to 1. Conversely, if the object is of the required type, the attention suppression ensures that the output stays close to zero. Although it might not be intuitive for humans, it efficiently detects whether the considered object is the one the sentence refers to.
- When the sentence refers to a navigation goal (e.g. *go top*, the attention highlights the agent's position (here y).
- When the sentence is a *grow* goal, the reward function focuses on the difference in object's size, while the policy further highlights the object's position.

The attention vectors uses information about the goal to highlight or suppress parts of the input using the different strategies described above depending on the type of input

(object categories, agent's position, difference in size etc). This type of gated-attention improves the interpretability of the reward function and policy.

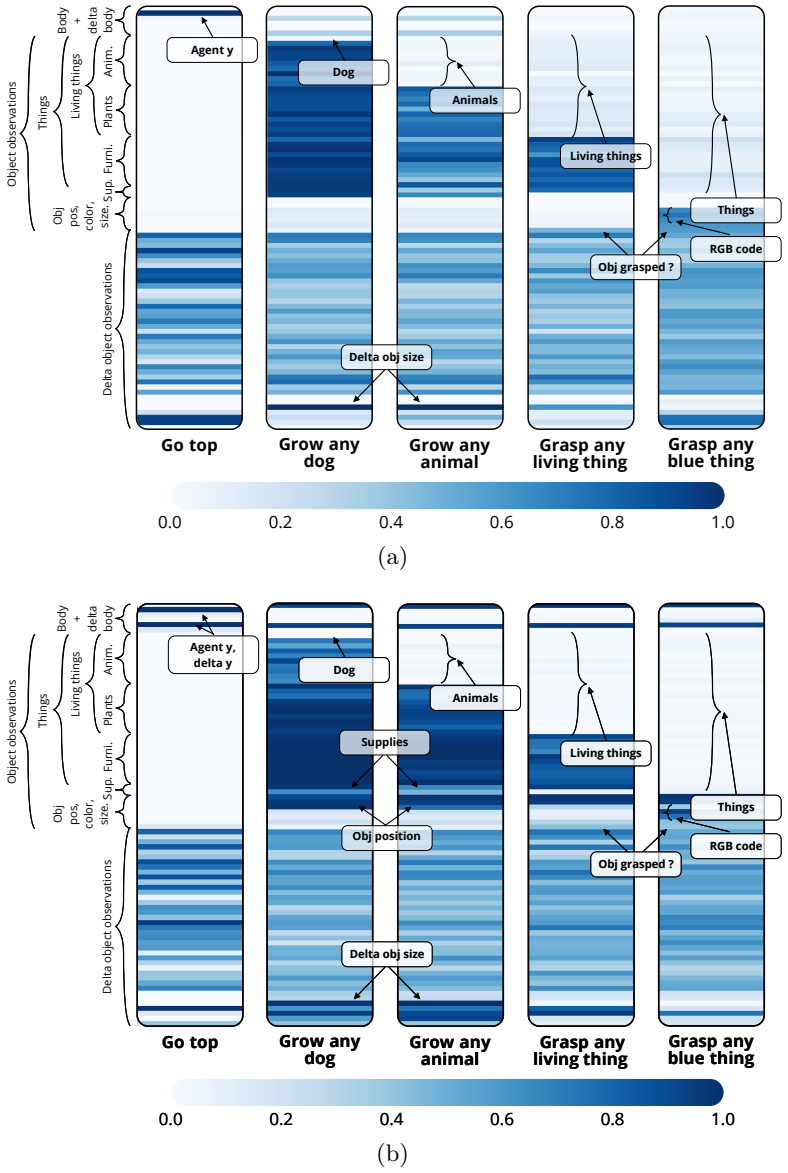


Figure D.10: **Attention vectors** (a) $\boldsymbol{\alpha}^g$ for the reward function (1 seed). (b) $\boldsymbol{\beta}^g$ for the policy (1 seed).

D.8 Comparing IMAGINE to Goal-as-state Approaches

In the goal-conditioned RL literature, some works have proposed goal generation mechanisms to facilitate the acquisition of skills over large sets of goals (Nair et al., 2018b; Pong et al., 2020; Colas et al., 2019a; Nair et al., 2020). Some of them had a special interest in exploration, and proposed to bias goal sampling towards goals from low density areas (Pong et al., 2020). One might then think that IMAGINE should be compared to these approaches. However, there are a few catches:

1. Nair et al. (2018b, 2020); Pong et al. (2020) use generative models of states to sample state-based goals. However, our environment is procedurally generated. This means that sampling a given state from the generative model has a very low probability to *match* the scene. If the present objects are three red cats, the agent has no chance to reach a goal specifying dogs and lions' positions, colors and sizes. Indeed, most of the state space is made of object features that cannot be acted upon (colors, types, sizes of most objects). One could imagine using SP to organize the scene, but we would need to ask SP to find the three objects specified by the generated goal, in the exact colors (RGB codes) and size. Doing so, there would be no distracting object for agent to discover and learn about. A second option is to condition the goal generation on the scene as it is done in **?**. The question of whether it might work in procedurally-generated environments remains open.

2. Assuming a perfect goal generator that only samples valid goals that do not ask a change of object color or type, the agent would then need to bring each object to its target position and to grow objects to their very specific goal size. These goals are not the same as those targeted by IMAGINE, they are too specific. These approaches –like most goal-conditioned RL approaches– represent goals as particular states (e.g. block positions in manipulation tasks, visual states in navigation tasks) (Schaul et al., 2015; Andrychowicz et al., 2017a; Nair et al., 2018b; Pong et al., 2020; Colas et al., 2019a). In contrast, language-conditioned agents represent abstract goals, usually defined by specific constraints on states (e.g. *grow any plant* requires the size of at least one plant to increase) (Chan et al., 2019a; Jiang et al., 2019a; Cideron et al., 2020c). For this reason, *goal-as-state* and *abstract goal* approaches do not tackle the same problem. The first targets specific coordinates, and cannot be instructed to reach abstract goals, while the second are not trained to reach specific states.

For these reasons, we argue that the goal-conditioned approaches that use state-based goals cannot be easily or fairly compared to our approach IMAGINE.

D.9 Implementation Details

Reward function inputs and hyperparameter.

Supplementary Sec. D.5 details the architecture of the reward function. The following provides extra details about the inputs. The object-dependent sub-state $\mathbf{s}_{obj(i)}$ contains information about both the agent's body and the corresponding object i: $\mathbf{s}_{obj(i)} = [\mathbf{o}_{body}, \Delta\mathbf{o}_{body}, \mathbf{o}_{obj(i)}, \Delta\mathbf{o}_{obj(i)}]$ where $\mathbf{o}_{body}$ and $\mathbf{o}_{obj(i)}$ are body- and obj_i-dependent observations, and $\Delta\mathbf{o}^t_{body} = \mathbf{o}^t_{body} - \mathbf{o}^0_{body}$ and $\Delta\mathbf{o}^t_{obj(i)} = \mathbf{o}^t_{obj(i)} - \mathbf{o}^0_{obj(i)}$ measure the difference between the initial and current observations. The second input is the attention vector $\boldsymbol{\alpha}^g$ that is integrated with $\mathbf{s}_{obj(i)}$ through an Hadamard product to form the model input: $\mathbf{x}^g_i = \mathbf{s}_{obj(i)} \odot \boldsymbol{\alpha}^g$. This attention vector is a simple mapping from $\mathbf{g}$ to a vector of the size of $\mathbf{s}_{obj(i)}$ contained in $[0,1]^{size(\mathbf{s}_{obj(i)})}$. This cast is implemented by a one-layer neural network with sigmoid activations $\texttt{NN}^{\text{cast}}$ such that $\boldsymbol{\alpha}^g = \texttt{NN}^{\text{cast}}(\mathbf{g})$.

For the three architectures the number of hidden units of the `LSTM` and the sizes of the hidden layers of fully connected networks are fixed to 100. `NN` parameters are initialized using He initialization (He et al., 2015) and we use one-hot word encodings. The `LSTM` is implemented using `rnn.BasicLSTMCell` from tensorflow 1.15 based on Zaremba et al. (2014). The states are initially set to zero. The `LSTM`'s weights are initialized uniformly from $[-0.1, 0.1]$ and the biases initially set to zero. The `LSTM` use a *tanh* activation function whereas the `NN` are using ReLU activation functions in their hidden layers and sigmoids at there output.

Reward function training schedule

The architecture are trained via backpropagation using the Adam Optimizer (Kingma & Ba, 2015). The data is fed to the model in batches of 512 examples. Each batch is constructed so that it contains at least one instance of each goal description g_{NL} (goals discovered so far). We also use a modular buffer to impose a ratio of positive rewards of 0.2 for each description in each batch. When trained in parallel of the policy, the reward function is updated once every 1200 episodes. Each update corresponds to up to 100 training epochs (100 batches). We implement a stopping criterion based on the F_1-score computed from a held-out test set uniformly sampled from the last episodes (20% of the last 1200 episodes (2 epochs)). The update is stopped when the F_1-score on the held-out set does not improve for 10 consecutive training epochs.

RL implementation and hyperparameters

In the policy and critic architectures, we use hidden layers of size 256 and ReLU activations. Attention vectors are cast from goal embeddings using single-layer neural networks with sigmoid activations. We use the He initialization scheme for (He et al., 2015) and train them via backpropagation using the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) (Kingma & Ba, 2015).

Our learning algorithm is built on top of the OpenAI Baselines implementation of HER-DDPG.[1] We leverage a parallel implementation with 6 actors. Actors share the same policy and critic parameters but maintain their own memory and conduct their own updates independently. Updates are then summed to compute the next set of parameters broadcast to all actors. Each actor is updated for 50 epochs with batches of size 256 every 2 episodes of environment interactions. Using hindsight replay, we enforce a ratio $p = 0.5$ of transitions associated with positive rewards in each batch. We use the same hyperparameters as Plappert et al. (2018).

Computing resources

The RL experiments contain 8 conditions of 10 seeds each, and 4 conditions with 5 seeds (SP study). Each run leverages 6 cpus (6 actors) for about 36h for a total of 2.5 cpu years. Experiments presented in this paper requires machines with at least 6 cpu cores.

[1] The OpenAI Baselines implementation of HER-DDPG can be found at https://github.com/openai/baselines, our implementation can be found at https://sites.google.com/view/imagine-drl.[2]

Bibliography

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pp. 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015430. URL https://doi.org/10.1145/1015330.1015430.

Achiam, J. and Sastry, S. Surprise-based intrinsic motivation for deep reinforcement learning. ArXiv - abs/1703.01732, 2017.

Achiam, J., Edwards, H., Amodei, D., and Abbeel, P. Variational option discovery algorithms. ArXiv - abs/1807.10299, 2018.

Ahilan, S. and Dayan, P. Feudal multi-agent hierarchies for cooperative reinforcement learning. *arXiv preprint arXiv:1901.08492*, 2019.

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N. J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., and Yan, M. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. *ArXiv – abs/2204.01691*, 2022.

Akakzia, A., Colas, C., Oudeyer, P.-Y., Chetouani, M., and Sigaud, O. DECSTR: Learning goal-directed abstract behaviors using pre-verbal spatial predicates in intrinsically motivated agents. In *Proc. of ICLR*, 2021a.

Akakzia, A., Colas, C., Oudeyer, P.-Y., Chetouani, M., and Sigaud, O. Grounding Language to Autonomously-Acquired Skills via Goal Generation. In *ICLR 2021 - Ninth International Conference on Learning Representation*, Vienna / Virtual, Austria, May 2021c. URL https://hal.inria.fr/hal-03121146.

Akakzia, A., Colas, C., Oudeyer, P.-Y., Chetouani, M., and Sigaud, O. Grounding Language to Autonomously-Acquired Skills Via Goal Generation. *Proc. of ICLR*, 2021b.

Allen, C. and Bekoff, M. *Species of Mind: The Philosophy and Biology of Cognitive Ethology*. MIT Press, 1999.

Andreas, J. Measuring compositionality in representation learning. In *International Conference on Learning Representations*, volume abs/1902.07181, 2019. URL https://openreview.net/forum?id=HJz05o0qK7.

Andreas, J. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7556–7566, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.676. URL https://www.aclweb.org/anthology/2020.acl-main.676.

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 39–48. IEEE Computer Society, 2016.

Andrychowicz, M., Crow, D., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight experience replay. In *Proc. of NeurIPS*, pp. 5048–5058, 2017a.

Andrychowicz, M., Crow, D., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight Experience Replay. *Proc. of NeurIPS*, 2017b.

Arandjelovic, R. and Zisserman, A. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Arora, A., Kaffee, L.-A., and Augenstein, I. Probing Pre-Trained Language Models for Cross-Cultural Differences in Values. *ArXiv – abs/2203.13722*, 2022.

Arora, S. and Doshi, P. A survey of inverse reinforcement learning: Challenges, methods and progress. 297:103500, 2021. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2021.103500. URL https://www.sciencedirect.com/science/article/pii/S0004370221000515.

Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., Ogino, M., and Yoshida, C. Cognitive developmental robotics: A survey. *IEEE transactions on autonomous mental development*, 1(1):12–34, 2009.

Ashby, W. R. Principles of the self-organizing system. In Foerster, H. V. and Jr, G. W. Z. (eds.), *Principles of Self-Organization: Transactions of the University of Illinois Symposium*, pp. 255–278. Pergamon Press, 1962.

Atance, C. M. Future thinking in young children. *Current Directions in Psychological Science*, 17(4):295–298, 2008.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. 47(2):235–256, 2002. ISSN 1573-0565. doi: 10.1023/A:1013689704352. URL https://doi.org/10.1023/A:1013689704352.

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., and Blundell, C. Agent57: Outperforming the atari human benchmark. In *Proc. of ICML*, volume 119, pp. 507–517, 2020a.

Badia, A. P., Sprechmann, P., Vitvitskyi, A., Guo, D., Piot, B., Kapturowski, S., Tieleman, O., Arjovsky, M., Pritzel, A., Bolt, A., and Blundell, C. Never give up: Learning directed exploration strategies. In *Proc. of ICLR*, 2020b.

Bahdanau, D., Hill, F., Leike, J., Hughes, E., Hosseini, S. A., Kohli, P., and Grefenstette, E. Learning to understand goal specifications by modelling reward. In *Proc. of ICLR*, 2019a.

Bahdanau, D., Hill, F., Leike, J., Hughes, E., Kohli, P., and Grefenstette, E. Learning to understand goal specifications by modelling reward. In *Proc. of ICLR*, 2019b.

Bahdanau, D., Murty, S., Noukhovitch, M., Nguyen, T. H., de Vries, H., and Courville, A. C. Systematic generalization: What is required and can it be learned? In *Proc. of ICLR*, 2019c.

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkxpxJBKwS.

Bandura, A. and Walters, R. H. *Social learning theory*, volume 1. Englewood cliffs Prentice Hall, 1977.

Baranes, A. and Oudeyer, P.-Y. Proximo-distal competence based curiosity-driven exploration. In *Learning, in International Conference on Epigenetic Robotics, Italie. Citeseer*. Citeseer, 2009a.

Baranes, A. and Oudeyer, P.-Y. R-iac: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, 1(3):155–169, 2009b.

Baranes, A. and Oudeyer, P.-Y. Intrinsically motivated goal exploration for active motor learning in robots: A case study. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1766–1773. IEEE, 2010.

Baranes, A. and Oudeyer, P.-Y. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.

Barde, P., Karch, T., Nowrouzezahrai, D., Moulin-Frier, C., Pal, C., and Oudeyer, P.-Y. Learning to guide and to be guided in the architect-builder problem. In *Proc. of ICLR*, 2022. URL https://openreview.net/forum?id=swiyAeGzFhQ.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks, 2018.

Beer, R. D. A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, 72(1):173–215, 1995. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(94)00005-L. URL https://www.sciencedirect.com/science/article/pii/000437029400005L.

Bellemare, M. G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. Unifying count-based exploration and intrinsic motivation. In *Proc. of NeurIPS*, pp. 1471–1479, 2016.

Bellemare, M. G., Candido, S., Castro, P. S., Gong, J., Machado, M. C., Moitra, S., Ponda, S. S., and Wang, Z. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

Bellman, R. Dynamic programming. *Science*, 153(3731):34–37, 1966.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.

Berk, L. E. Why Children Talk to Themselves. *Scientific American*, 1994.

Berlyne, D. E. Curiosity and exploration. *Science*, 153(3731):25–33, 1966.

Berseth, G., Geng, D., Devin, C., Finn, C., Jayaraman, D., and Levine, S. Smirl: Surprise minimizing rl in dynamic environments. *arXiv preprint arXiv:1912.05510*, 2019.

Besse, P., Guillouet, B., Loubes, J.-M., and François, R. Review and perspective for distance based trajectory clustering, 2015.

Bisk, Y., Holtzman, A., Thomason, J., Andreas, J., Bengio, Y., Chai, J., Lapata, M., Lazaridou, A., May, J., Nisnevich, A., Pinto, N., and Turian, J. Experience grounds language. In *Proc. of EMNLP*. Association for Computational Linguistics, 2020.

Blaes, S., Pogancic, M. V., Zhu, J., and Martius, G. Control what you can: Intrinsically motivated task-planning agent. In *Proc. of NeurIPS*, pp. 12520–12531, 2019.

Bonawitz, E., Shafto, P., Gweon, H., Goodman, N., Spelke, E., and Schulz, L. The double-edged sword of pedagogy: Instruction limits spontaneous exploration and discovery. *Cognition*, 120:322–30, 09 2011.

Bornstein, M. H., Tamis-LeMonda, C. S., Tal, J., Ludemann, P., Toda, S., Rahn, C. W., Pêcheux, M.-G., Azuma, H., and Vardi, D. Maternal Responsiveness to Infants in Three Societies: The United States, France, and Japan. *Child development*, 63(4), 1992. Publisher: Wiley Online Library.

Boyd, B. The evolution of stories: from mimesis to language, from fact to fiction. *Wiley Interdisciplinary Reviews: Cognitive Science*, 9(1):e1444, 2018.

Boyd, R., Richerson, P. J., and Henrich, J. The cultural niche: Why social learning is essential for human adaptation. *Proceedings of the National Academy of Sciences*, 108 (supplement_2):10918–10925, 2011.

Branavan, S., Zettlemoyer, L., and Barzilay, R. Reading between the lines: Learning to map high-level instructions to commands. In *Proc. of ACL*, pp. 1268–1277. Association for Computational Linguistics, 2010.

Brewer, K., Pollock, N., and Wright, F. V. Addressing the Challenges of Collaborative Goal Setting with Children and Their Families. *Physical & Occupational Therapy in Pediatrics*, 2014.

Brighton, H. Compositional syntax from cultural transmission. *Artificial Life*, 8:25–54, 2002.

Brighton, H. and Kirby, S. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial Life*, 12:229–242, 2006.

Brown, D. S., Goo, W., and Niekum, S. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In Kaelbling, L. P., Kragic, D., and Sugiura, K. (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 330–359. PMLR, 30 Oct–01 Nov 2020a. URL https://proceedings.mlr.press/v100/brown20a.html.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. *Proc. of NeurIPS*, abs/2005.14165, 2020b.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

Bruner, J. Child's talk: Learning to use language. *Child Language Teaching and Therapy*, 1(1):111–114, 1985.

Bruner, J. *Acts of meaning*. Harvard university press, 1990.

Bruner, J. The narrative construction of reality. *Critical inquiry*, 18(1):1–21, 1991.

Burda, Y., Edwards, H., Storkey, A. J., and Klimov, O. Exploration by random network distillation. In *Proc. of ICLR*, 2019.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. Monet: Unsupervised scene decomposition and representation. *ArXiv - abs/1901.11390*, 2019.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraula, G., and Bonabeau, E. *Self-Organization in Biological Systems*. Princeton University Press, Princeton, 2001. ISBN 9780691212920. doi: doi:10.1515/9780691212920. URL https://doi.org/10.1515/9780691212920.

Campero, A., Raileanu, R., Küttler, H., Tenenbaum, J. B., Rocktäschel, T., and Grefenstette, E. Learning with AMIGo: Adversarially Motivated Intrinsic Goals. *Proc. of ICLR*, 2021.

Cangelosi, A. and Parisi, D. Simulating the evolution of language. In *Springer London*, 2002.

Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., Tani, J., Belpaeme, T., Sandini, G., Nori, F., et al. Integration of action and language knowledge: A roadmap for developmental robotics. *IEEE Transactions on Autonomous Mental Development*, 2(3), 2010a.

Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., Tani, J., Belpaeme, T., Sandini, G., Nori, F., Fadiga, L., Wrede, B., Rohlfing, K., Tuci, E., Dautenhahn, K., Saunders, J., and Zeschel, A. Integration of action and language knowledge: A roadmap for developmental robotics. *IEEE Transactions on Autonomous Mental Development*, 2(3):167–195, 2010b. doi: 10.1109/TAMD.2010.2053034.

Cao, K., Lazaridou, A., Lanctot, M., Leibo, J. Z., Tuyls, K., and Clark, S. Emergent communication through negotiation. In *International Conference on Learning Representations*, 2018.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers, 2020.

Carruthers, P. Thinking in language?: Evolution and a modularist possibility. In *Language and Thought*. Cambridge University Press, 1998.

Carruthers, P. Modularity, Language, and the Flexibility of Thought. *Behavioral and Brain Sciences*, (6), 2002. ISSN 0140-525X, 1469-1825.

Carruthers, P. and Boucher, J. *Language and Thought*. Cambridge University Press, 1998.

Carta, T., Oudeyer, P.-Y., Sigaud, O., and Lamprier, S. Eager: Asking and answering questions for automatic reward shaping in language-guided rl, 2022.

Carta, T., Romac, C., Wolf, T., Lamprier, S., Sigaud, O., and Oudeyer, P.-Y. Grounding large language models in interactive environments with online reinforcement learning, 2023.

Cederborg, T. and Oudeyer, P.-Y. A social learning formalism for learners trying to figure out what a teacher wants them to do. *Paladyn: Journal of Behavioral Robotics*, 5:64–99, 2014.

Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., and Baroni, M. Compositionality and generalization in emergent languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4427–4442, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.407. URL https://aclanthology.org/2020.acl-main.407.

Chaabouni, R., Kharitonov, E., Dupoux, E., and Baroni, M. Communicating artificial neural networks develop efficient color-naming systems. *Proceedings of the National Academy of Sciences of the United States of America*, 118, 2021.

Chaabouni, R., Strub, F., Altché, F., Tarassov, E., Tallec, C., Davoodi, E., Mathewson, K. W., Tieleman, O., Lazaridou, A., and Piot, B. Emergent communication at scale. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=AUGBfDIV9rL.

Chalmers, D. J. Connectionism and compositionality: Why fodor and pylyshyn were wrong. *Philosophical Psychology*, 6(3):305–319, 1993. doi: 10.1080/09515089308573094.

Chan, H., Wu, Y., Kiros, J., Fidler, S., and Ba, J. Actrce: Augmenting experience via teacher's advice for multi-goal reinforcement learning. ArXiv - abs/1902.04546, 2019a.

Chan, H., Wu, Y., Kiros, J., Fidler, S., and Ba, J. ACTRCE: Augmenting Experience via Teacher's Advice For Multi-Goal Reinforcement Learning. *ArXiv – abs/1902.04546*, 2019b.

Chaplot, D. S., Sathyendra, K. M., Pasumarthi, R. K., Rajagopal, D., and Salakhutdinov, R. Gated-Attention Architectures for Task-Oriented Language Grounding. *Proc. of AAAI*, 2018a.

Chaplot, D. S., Sathyendra, K. M., Pasumarthi, R. K., Rajagopal, D., and Salakhutdinov, R. Gated-attention architectures for task-oriented language grounding. In *Proc. of AAAI*, pp. 2819–2826, 2018b.

Chen, D. L. and Mooney, R. J. Learning to interpret natural language navigation instructions from observations. In *Proc. of AAAI*, 2011a.

Chen, D. L. and Mooney, R. J. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pp. 859–865. AAAI Press, 2011b.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Chen, V., Gupta, A., and Marino, K. Ask Your Humans: Using Human Instructions to Improve Generalization in Reinforcement Learning. *Proc. of ICLR*, 2021.

Cheney, D. L. and Seyfarth, R. M. Constraints and preadaptations in the earliest stages of language evolution. 2005.

Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T. H., and Bengio, Y. BabyAI: First Steps Towards Grounded Language Learning With a Human In the Loop. In *Proc. of ICLR*, 2019a.

Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T. H., and Bengio, Y. Baby-Ai: First Steps Towards Grounded Language Learning with a Human in the Loop. *Proc. of ICLR*, 2019b.

Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T. H., and Bengio, Y. Babyai: A platform to study the sample efficiency of grounded language learning, 2019c.

Chiang, K.-J., Emmanouilidou, D., Gamper, H., Johnston, D., Jalobeanu, M., Cutrell, E., Wilson, A., An, W. W., and Tashev, I. A closed-loop adaptive brain-computer interface framework: Improving the classifier with the use of error-related potentials. In *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 487–490, 2021. doi: 10.1109/NER49283.2021.9441133.

Choi, E., Lazaridou, A., and de Freitas, N. Compositional obverter communication learning from raw visual input. In *Proc. of ICLR*, 2018a.

Choi, E., Lazaridou, A., and de Freitas, N. Multi-agent compositional communication learning from raw visual input. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=rknt2Be0-.

Choi, J., Sharma, A., Lee, H., Levine, S., and Gu, S. S. Variational Empowerment as Representation Learning for Goal-Based Reinforcement Learning. ArXiv - abs/2106.01404, 2021.

Chomsky, N. *Syntactic structures*. Mouton, 1957a. ISBN 9789027933850.

Chomsky, N. *Syntactic Structures*. Mouton, 1957b. ISBN 978-90-279-3385-0.

Chomsky, N. *Reflections on Language*. Number v. 10 in Pantheon Books. Pantheon Books, 1975. ISBN 9780394499567. URL https://books.google.fr/books?id=R78kAQAAMAAJ.

Chopra, S., Tessler, M. H., and Goodman, N. D. The first crank of the cultural ratchet: Learning and transmitting concepts through language. In *CogSci*, pp. 226–232, 2019.

Chu, J. and Schulz, L. Exploratory play, rational action, and efficient search. In Denison, S., Mack, M., 0023, Y. X., and Armstrong, B. C. (eds.), *Proceedings of the 42th Annual Meeting of the Cognitive Science Society - Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020, virtual, July 29 - August 1, 2020*. cognitivesciencesociety.org, 2020. URL https://cogsci.mindmodeling.org/2020/papers/0169/index.html.

Cideron, G., Seurin, M., Strub, F., and Pietquin, O. Higher: Improving instruction following with hindsight generation for experience replay. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 225–232, 2020a. doi: 10.1109/SSCI47803.2020.9308603.

Cideron, G., Seurin, M., Strub, F., and Pietquin, O. HIGhER: Improving Instruction Following with Hindsight Generation for Experience Replay. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020b.

Cideron, G., Seurin, M., Strub, F., and Pietquin, O. Higher: Improving instruction following with hindsight generation for experience replay. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 225–232. IEEE, 2020c.

Clark, A. Magic Words: How Language Augments Human Computation. In Carruthers, P. and Boucher, J. (eds.), *Language and Thought*. Cambridge University Press, 1 edition, 1998. ISBN 978-0-521-63108-2 978-0-521-63758-9 978-0-511-59790-9.

Co-Reyes, J. D., Gupta, A., Sanjeev, S., Altieri, N., Andreas, J., DeNero, J., Abbeel, P., and Levine, S. Guiding policies with language via meta-learning. In *Proc. of ICLR*, 2019.

Codevilla, F., Müller, M., López, A., Koltun, V., and Dosovitskiy, A. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9. IEEE, 2018.

Colas, C., Oudeyer, P., Sigaud, O., Fournier, P., and Chetouani, M. CURIOUS: intrinsically motivated modular multi-goal reinforcement learning. In *Proc. of ICML*, volume 97, pp. 1331–1340, 2019a.

Colas, C., Sigaud, O., and Oudeyer, P.-Y. A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms. *ArXiv - abs/1904.06979*, 2019b.

Colas, C., Akakzia, A., Oudeyer, P.-Y., Chetouani, M., and Sigaud, O. Language-conditioned goal generation: a new approach to language grounding for rl. ArXiv - abs/2006.07043, 2020a.

Colas, C., Karch, T., Lair, N., Dussoux, J., Moulin-Frier, C., Dominey, P. F., and Oudeyer, P. Language as a cognitive tool to imagine goals in curiosity driven exploration. In *Proc. of NeurIPS*, 2020b.

Colas, C., Karch, T., Moulin-Frier, C., and Oudeyer, P.-Y. Language and Culture Internalisation for Human-Like Autotelic AI. *Nature Machine Intelligence*, 2022a.

Colas, C., Karch, T., Sigaud, O., and Oudeyer, P.-Y. Autotelic Agents with Intrinsically Motivated Goal-conditioned Reinforcement Learning: a Short Survey. *Journal of Artificial Intelligence Research*, 2022b.

Crawford, V. P. and Sobel, J. Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pp. 1431–1451, 1982.

Creswell, A., Nikiforou, K., Vinyals, O., Saraiva, A., Kabra, R., Matthey, L., Burgess, C., Reynolds, M., Tanburn, R., Garnelo, M., and Shanahan, M. Alignnet: Unsupervised entity alignment, 2020.

Csikzentmihalyi, M. Finding flow: The psychology of engagement with everyday life. *New York: Basic*, 1997.

Cummins, R. Systematicity. *The Journal of Philosophy*, 93(12), 1996. Publisher: JSTOR.

Côté, M.-A., K\'{a}d\'{a}r, \., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Moore, J., Hausknecht, M. J., Asri, L. E., Adada, M., Tay, W., and Trischler, A. TextWorld: A Learning Environment for Text-Based Games. *Computer Games - 7th Workshop at IJCAI*, 2018.

Dai, S., Xu, W., Hofmann, A., and Williams, B. An Empowerment-based Solution to Robotic Manipulation Tasks with Sparse Rewards. ArXiv - abs/2010.07986, 2020.

Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., and Batra, D. Embodied question answering. *Proc. of CVPR*, 2018.

Dautenhahn, K. and Billard, A. Studying Robot Social Cognition Within a Developmental Psychology Framework. *Proc. of Eurobot (IEEE)*, 1999.

Dayan, P. and Hinton, G. E. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pp. 271–278, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. ISBN 1558602747.

Dayan, P. and Hinton, G. E. Feudal reinforcement learning. In *Proc. of NeurIPS*, pp. 271–278, 1993a.

Dayan, P. and Hinton, G. E. Feudal Reinforcement Learning. In *Advances in neural information processing systems*, 1993b.

de Boer, B. G. Self-organization in vowel systems. *J. Phonetics*, 28:441–465, 2000.

deBettencourt, M. T., Cohen, J. D., Lee, R. F., Norman, K. A., and Turk-Browne, N. B. Closed-loop training of attention with real-time brain imaging. *Nature neuroscience*, 18:470 – 475, 2015.

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Dennett, D. C. *Consciousness Explained*. Penguin uk, 1993.

Dessalles, J.-L. *Aux origines du langage*. Hermès-science, 2000.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Dilts, R. Nlp and self-organization theory. *Anchor Point*, 9(6):14–21, 1995.

Ding, D., Hill, F., Santoro, A., and Botvinick, M. Object-based attention for spatio-temporal reasoning: Outperforming neuro-symbolic models with flexible distributed architectures, 2020.

Ding, Y., Florensa, C., Abbeel, P., and Phielipp, M. Goal-conditioned imitation learning. In *Proc. of NeurIPS*, pp. 15298–15309, 2019.

Dominey, P. F. Emergence of grammatical constructions: evidence from simulation and grounded agent experiments. *Connection Science*, 17(3-4):289–306, 2005. ISSN 0954-0091.

Dove, G. Language as a Disruptive Technology: Abstract Concepts, Embodiment and the Flexible Mind. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2018.

Du, Y., Liu, B., Moens, V., Liu, Z., Ren, Z., Wang, J., Chen, X., and Zhang, H. Learning correlated communication topology in multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, pp. 456–464, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.

Eccles, T., Bachrach, Y., Lever, G., Lazaridou, A., and Graepel, T. Biases for emergent communication in multi-agent reinforcement learning. In *NeurIPS*, volume 32, 2019.

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., and Clune, J. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Elliot, A. J. and Fryer, J. W. The goal construct in psychology. *Handbook of motivation science*, 18:235–250, 2008.

Elliott, E. M., Morey, C. C., AuBuchon, A. M., Cowan, N., Jarrold, C., Adams, E. J., Attwood, M., Bayram, B., Beeler-Duden, S., Blakstvedt, T. Y., Büttner, G., Castelain, T., Cave, S., Crepaldi, D., Fredriksen, E., Glass, B. A., Graves, A. J., Guitard, D., Hoehl, S., Hosch, A., Jeanneret, S., Joseph, T. N., Koch, C., Lelonkiewicz, J. R., Lupyan, G., McDonald, A., Meissner, G., Mendenhall, W., Moreau, D., Ostermann, T., Özdoğru, A. A., Padovani, F., Poloczek, S., Röer, J. P., Schonberg, C. C., Tamnes, C. K., Tomasik, M. J., Valentini, B., Vergauwe, E., Vlach, H. A., and Voracek, M. Multilab Direct Replication of Flavell, Beach, and Chinsky (1966): Spontaneous Verbal Rehearsal in a Memory Task as a Function of Age. *Advances in Methods and Practices in Psychological Science*, 4(2), 2021.

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. Genesis: Generative scene inference and sampling with object-centric latent representations, 2020.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. In *Proc. of ICLR*, 2019.

Eysenbach, B., Geng, X., Levine, S., and Salakhutdinov, R. R. Rewriting history with inverse RL: hindsight inference for policy improvement. In *Proc. of NeurIPS*, 2020.

Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Proc. of AAAI*, 2022.

Fang, K., Zhu, Y., Savarese, S., and Fei-Fei, L. Discovering Generalizable Skills via Automated Generation of Diverse Tasks. In *Proceedings of Robotics: Science and Systems*, 2021.

Ferreira, M., Conceição, H., Viriyasitavat, W., and Tonguz, O. Self-organized traffic control. pp. 85–90, 09 2010. doi: 10.1145/1860058.1860077.

Florensa, C., Held, D., Geng, X., and Abbeel, P. Automatic goal generation for reinforcement learning agents. In *Proc. of ICML*, volume 80, pp. 1514–1523, 2018.

Florensa, C., Degrave, J., Heess, N., Springenberg, J. T., and Riedmiller, M. Self-supervised learning of image embedding for continuous control. ArXiv - abs/1901.00943, 2019.

Fodor, J. A. *The Language of Thought*. Harvard university press, 1975.

Fodor, J. A. and Pylyshyn, Z. W. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988a. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(88)90031-5. URL https://www.sciencedirect.com/science/article/pii/0010027788900315.

Fodor, J. A. and Pylyshyn, Z. W. Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition*, 28(1-2), 1988b. Publisher: Elsevier.

Foerster, J. N., Assael, Y., de Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In *Proc. of NeurIPS*, 2016.

Forestier, S. and Oudeyer, P.-Y. Modular active curiosity-driven discovery of tool use. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 3965–3972. IEEE, 2016.

Forestier, S., Portelas, R., Mollard, Y., and Oudeyer, P.-Y. Intrinsically motivated goal exploration processes with automatic curriculum learning. *Journal of Machine Learning Research*, 23(152):1–41, 2022. URL http://jmlr.org/papers/v23/21-0808.html.

Fournier, P., Sigaud, O., Chetouani, M., and Oudeyer, P.-Y. Accuracy-based curriculum learning in deep reinforcement learning. ArXiv - abs/1806.09614, 2018.

Fournier, P., Colas, C., Chetouani, M., and Sigaud, O. Clic: Curriculum learning and imitation for object control in nonrewarding environments. *IEEE Transactions on Cognitive and Developmental Systems*, 13(2):239–248, 2021. doi: 10.1109/TCDS.2019.2933371.

Frans, K., Ho, J., Chen, X., Abbeel, P., and Schulman, J. Meta learning shared hierarchies. In *Proc. of ICLR*, 2018.

Frege, G. *Philosophical and mathematical correspondence*. Blackwell, 1980.

Freire, I. T., Moulin-Frier, C., Sanchez-Fibla, M., Arsiwalla, X. D., and Verschure, P. F. M. J. Modeling the formation of social conventions from embodied real-time interactions. *PLOS ONE*, 15(6):1–22, 06 2020. doi: 10.1371/journal.pone.0234434. URL https://doi.org/10.1371/journal.pone.0234434.

Fu, J., Korattikara, A., Levine, S., and Guadarrama, S. From language to goals: Inverse reinforcement learning for vision-based instruction following. In *Proc. of ICLR*, 2019.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *Proc. of ICML*, volume 97, pp. 2052–2062, 2019.

Galantucci, B. and Garrod, S. Experimental semiotics: a review. *Frontiers in human neuroscience*, 5:11, 2011.

Gentner, D. and Hoyos, C. Analogy and Abstraction. *Topics in Cognitive Science*, (3), 2017. ISSN 17568757.

Gentner, D. and Loewenstein, J. *Relational Language and Relational Thought*. Erlbaum, 2002.

Gibson, J. J. J. J. *The senses considered as perceptual systems*. Allen & Unwin, London, 1968.

Glenberg, A. M. and Kaschak, M. P. Grounding language in action. *Psychonomic Bulletin & Review*, 9(3):558–565, 2002a. ISSN 1069-9384.

Glenberg, A. M. and Kaschak, M. P. Grounding language in action. *Psychonomic Bulletin & Review*, 9(3):558–565, September 2002b. ISSN 1531-5320. doi: 10.3758/BF03196313. URL https://doi.org/10.3758/BF03196313.

Goldberg, A. E. The Emergence of the Semantics of Argument Structure Constructions. In *The emergence of language*. Psychology Press, 1999.

Goldberg, A. E. Constructions: A new theoretical approach to language. *Trends in cognitive sciences*, 7(5):219–224, 2003.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *Proc. of NeurIPS*, pp. 2672–2680, 2014.

Goodrich, M. A. and Schultz, A. C. *Human-robot interaction: a survey*. Now Publishers Inc, 2008.

Gopnik, A., Meltzoff, A. N., and Kuhl, P. K. *The scientist in the crib: Minds, brains, and how children learn*. William Morrow & Co, 1999.

Gottlieb, J. and Oudeyer, P.-Y. Towards a neuroscience of active sampling and curiosity. *Nature Reviews Neuroscience*, 19(12):758–770, 2018.

Goyal, P., Niekum, S., and Mooney, R. J. Using natural language for reward shaping in reinforcement learning. In *Proc. of IJCAI*, pp. 2385–2391, 2019.

Green, E. J. and Quilty-Dunn, J. What is an object file? *The British Journal for the Philosophy of Science*, 2017.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference. In *Proc. of ICML*, volume 97, pp. 2424–2433, 2019.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-object representation learning with iterative variational inference, 2020.

Gregor, K., Rezende, D. J., and Wierstra, D. Variational intrinsic control. ArXiv - abs/1611.07507, 2016.

Grizou, J., Lopes, M., and Oudeyer, P.-Y. Robot learning simultaneously a task and how to interpret human instructions. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pp. 1–8, 2013. doi: 10.1109/DevLrn.2013.6652523.

Grizou, J., Iturrate, I., Montesano, L., Oudeyer, P.-Y., and Lopes, M. Calibration-Free BCI Based Control. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1–8, Quebec, Canada, July 2014. URL https://hal.archives-ouvertes.fr/hal-00984068.

Gupta, A., Resnick, C., Foerster, J., Dai, A., and Cho, K. Compositionality and capacity in emergent languages. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 34–38, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.5. URL https://aclanthology.org/2020.repl4nlp-1.5.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Hadfield-Menell, D., Russell, S. J., Abbeel, P., and Dragan, A. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29:3909–3917, 2016.

Hannafin, M. J., Hall, C., Land, S., and Hill, J. Learning in open-ended environments: Assumptions, methods, and implications. *Educational Technology*, 34(8):48–55, 1994. ISSN 00131962. URL http://www.jstor.org/stable/44428230.

Harari, Y. N. *Sapiens: A Brief History of Humankind*. Random House, 2014.

Hartikainen, K., Geng, X., Haarnoja, T., and Levine, S. Dynamical distance learning for semi-supervised and unsupervised skill discovery. In *Proc. of ICLR*, 2020.

Havrylov, S. and Titov, I. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/70222949cc0db89ab32c9969754d4758-Paper.pdf.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Henrich, J. and McElreath, R. The evolution of cultural evolution. *Evolutionary Anthropology: Issues, News, and Reviews.*, 2003.

Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W. M., Jaderberg, M., Teplyashin, D., Wainwright, M., Apps, C., Hassabis, D., and Blunsom, P. Grounded Language Learning in a Simulated 3D World. ArXiv - abs/1706.06551, 2017a.

Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W. M., Jaderberg, M., Teplyashin, D., Wainwright, M., Apps, C., Hassabis, D., and Blunsom, P. Grounded Language Learning in a Simulated 3D World. *ArXiv – abs/1706.06551*, 2017b.

Hermer-Vazquez, L. Language, Space, and the Development of Cognitive Flexibility in Humans: The Case of Two Spatial Memory Tasks. *Cognition*, (3), 2001. ISSN 00100277.

Hershcovich, D., Frank, S., Lent, H., de Lhoneux, M., Abdou, M., Brandl, S., Bugliarello, E., Piqueras, L. C., Chalkidis, I., Cui, R., Fierro, C., Margatina, K., Rust, P., and Søgaard, A. Challenges and Strategies in Cross-Cultural NLP. *Proc. of ACL*, 2022.

Hesse, M. The Cognitive Claims of Metaphor. *The journal of speculative philosophy*, 1988.

Hill, F., Lampinen, A., Schneider, R., Clark, S., Botvinick, M., McClelland, J. L., and Santoro, A. Environmental drivers of systematicity and generalization in a situated agent. In *Proc. of ICLR*, 2020a.

Hill, F., Mokra, S., Wong, N., and Harley, T. Human Instruction-Following with Deep Reinforcement Learning via Transfer-Learning from Text. *ArXiv – abs/2005.09382*, 2020b.

Hinaut, X. and Dominey, P. F. Real-time parallel processing of grammatical structure in the fronto-striatal system: A recurrent network simulation study using reservoir computing. *PloS one*, 8(2), 2013.

Hinaut, X., Petit, M., Pointeau, G., and Dominey, P. F. Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks. *Frontiers in neurorobotics*, 8:16, 2014.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Hockett, C. F. and Hockett, C. D. The origin of speech. *Scientific American*, 203(3): 88–97, 1960.

Hoffmann, T. Construction Grammar and Creativity: Evolution, Psychology, and Cognitive Science. *Cognitive Semiotics*, (1), 2020. ISSN 2235-2066, 1662-1425.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., Turck, F. D., and Abbeel, P. VIME: variational information maximizing exploration. In *Proc. of NeurIPS*, pp. 1109–1117, 2016.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *Proc. of ICML*, 2022.

Hui, D. Y.-T., Chevalier-Boisvert, M., Bahdanau, D., and Bengio, Y. Babyai 1.1, 2020.

Hunt, J. M. Intrinsic motivation and its role in psychological development. *Nebraska symposium on motivation*, 13:189–282, 1965. URL https://cir.nii.ac.jp/crid/1571698599234799104.

Hupkes, D., Dankers, V., Mul, M., and Bruni, E. Compositionality decomposed: how do neural networks generalise?, 2020.

Hurford, J. R. Biological evolution of the saussurean sign as a component of the language acquisition device. *Lingua*, 77:187–222, 1989.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P. A., Strouse, D. J., Leibo, J., and de Freitas, N. Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning. *Proc. of ICML*, 2019.

Jiang, J. and Lu, Z. Learning attentional communication for multi-agent cooperation. In *NeurIPS*, 2018.

Jiang, Y., Gu, S., Murphy, K., and Finn, C. Language as an abstraction for hierarchical deep reinforcement learning. In *Proc. of NeurIPS*, pp. 9414–9426, 2019a.

Jiang, Y., Gu, S., Murphy, K., and Finn, C. Language as an abstraction for hierarchical deep reinforcement learning. *Proc. of NeurIPS*, 2019b.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning, 2016.

Johnson, S. P., Amso, D., and Slemmer, J. A. Development of object concepts in infancy: Evidence for early learning in an eye-tracking paradigm. *Proceedings of the National Academy of Sciences*, 100(18):10568–10573, 2003.

Kaelbling, L. P. Learning to achieve goals. In *IJCAI*, pp. 1094–1099. Citeseer, 1993.

Kalinowska, A., Davoodi, E., Strub, F., Mathewson, K., Murphey, T., and Pilarski, P. Towards situated communication in multi-step interactions: Time is a key pressure in communication emergence. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 44, 2022.

Kaplan, F. and Oudeyer, P.-Y. In search of the neural circuits of intrinsic motivation. *Frontiers in neuroscience*, 1:17, 2007.

Karch, T., Colas, C., Teodorescu, L., Moulin-Frier, C., and Oudeyer, P.-Y. Deep sets for generalization in rl, 2020. URL https://arxiv.org/abs/2003.09443.

Karch, T., Teodorescu, L., Hofmann, K., Moulin-Frier, C., and Oudeyer, P.-Y. Grounding Spatio-Temporal Language with Transformers. *Proc. of NeurIPS*, 2021.

Karch, T., Lemesle, Y., Laroche, R., Moulin-Frier, C., and Oudeyer, P.-Y. Contrastive multimodal learning for emergence of graphical sensory-motor communication, 2023. URL https://arxiv.org/abs/2210.06468.

Katyal, K. D., Johannes, M. S., Kellis, S., Aflalo, T., Klaes, C., McGee, T. G., Para, M. P., Shi, Y., Lee, B., Pejsa, K., Liu, C., Wester, B. A., Tenore, F., Beaty, J. D., Ravitz, A. D., Andersen, R. A., and McLoughlin, M. P. A collaborative bci approach to autonomous control of a prosthetic limb system. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1479–1482, 2014. doi: 10.1109/SMC.2014.6974124.

Ke, L., Choudhury, S., Barnes, M., Sun, W., Lee, G., and Srinivasa, S. Imitation learning as f-divergence minimization, 2020.

Kidd, C. and Hayden, B. Y. The psychology and neuroscience of curiosity. *Neuron*, 88 (3):449–460, 2015a.

Kidd, C. and Hayden, B. Y. The Psychology and Neuroscience of Curiosity. *Neuron*, 2015b.

Kidd, C., Piantadosi, S. T., and Aslin, R. N. The goldilocks effect: Human infants allocate attention to visual sequences that are neither too simple nor too complex. *PLoS ONE*, 7, 2012.

Kim, K., Sano, M., Freitas, J. D., Haber, N., and Yamins, D. Active world model learning with progress curiosity. In *Proc. of ICML*, volume 119, pp. 5306–5315, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017.

Kirby, S. Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity. *IEEE Trans. Evol. Comput.*, 5:102–110, 2001.

Kirby, S., Griffiths, T., and Smith, K. Iterated learning and the evolution of language. *Current Opinion in Neurobiology*, 28:108–114, 2014.

Kiseleva, J., Li, Z., Aliannejadi, M., Mohanty, S., ter Hoeve, M., Burtsev, M., Skrynnik, A., Zholus, A., Panov, A., Srinet, K., et al. Neurips 2021 competition iglu: Interactive grounded language understanding in a collaborative environment. *arXiv preprint arXiv:2110.06536*, 2021.

Klein, E., Geist, M., and Pietquin, O. Batch, Off-policy and Model-free Apprenticeship Learning. In *EWRL 2011*, pp. 1–12, Athens, Greece, September 2011. URL https://hal-supelec.archives-ouvertes.fr/hal-00660623.

Kottur, S., Moura, J. M. F., Lee, S., and Batra, D. Natural language does not emerge 'naturally' in multi-agent dialog. In *EMNLP*, 2017.

Kovač, G., Laversanne-Finot, A., and Oudeyer, P.-Y. Grimgep: Learning progress for robust goal sampling in visual deep reinforcement learning. ArXiv - abs/2008.04388, 2020.

Kulkarni, T. D., Narasimhan, K., Saeedi, A., and Tenenbaum, J. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Proc. of NeurIPS*, pp. 3675–3683, 2016.

Lake, B. M. and Baroni, M. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks, 2018.

Lakoff, G. and Johnson, M. *Metaphors We Live By*. University of Chicago press, 2008.

Lampinen, A. K., Roy, N. A., Dasgupta, I., Chan, S. C. Y., Tam, A. C., McClelland, J. L., Yan, C., Santoro, A., Rabinowitz, N. C., Wang, J. X., and Hill, F. Tell Me Why! – Explanations Support Learning of Relational and Causal Structure. *Proc. of ICML*, 2022.

Lanier, J. B., McAleer, S., and Baldi, P. Curiosity-driven multi-criteria hindsight experience replay. ArXiv - abs/1906.03710, 2019.

Laskin, M., Wang, L., Oh, J., Parisotto, E., Spencer, S., Steigerwald, R., Strouse, D., Hansen, S., Filos, A., Brooks, E., Gazeau, M., Sahni, H., Singh, S., and Mnih, V. In-context reinforcement learning with algorithm distillation, 2022. URL https://arxiv.org/abs/2210.14215.

Laversanne-Finot, A., Pere, A., and Oudeyer, P.-Y. Curiosity driven exploration of learned disentangled goal spaces. In *Conference on Robot Learning*, pp. 487–504. PMLR, 2018.

Lazaridou, A., Peysakhovich, A., and Baroni, M. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hk8N3Sclg.

Lazaridou, A., Hermann, K. M., Tuyls, K., and Clark, S. Emergence of linguistic communication from referential games with symbolic and pixel input. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJGv1Z-AW.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324, 1998.

Levy, A., Platt, R., and Saenko, K. Hierarchical reinforcement learning with hindsight. ArXiv - abs/1805.08180, 2018.

Lewis, D. K. *Convention: A Philosophical Study*. Cambridge, MA, USA: Wiley-Blackwell, 1969.

Li, F. and Bowling, M. Ease-of-teaching and language structure from emergent communication. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/b0cf188d74589db9b23d5d277238a929-Paper.pdf.

Li, R., Jabri, A., Darrell, T., and Agrawal, P. Towards practical multi-object manipulation using relational reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4051–4058. IEEE, 2020.

Liang, P. P., Wu, C., Morency, L.-P., and Salakhutdinov, R. Towards understanding and mitigating social biases in language models. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6565–6576. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liang21a.html.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *Proc. of ICLR*, 2016.

Lindblom, J. and Ziemke, T. Social Situatedness of Natural and Artificial Intelligence: Vygotsky and Beyond. *Adaptive Behavior*, (2), 2003. ISSN 1059-7123, 1741-2633.

Linke, C., Ady, N. M., White, M., Degris, T., and White, A. Adapting behavior via intrinsic reward: a survey and empirical study. *Journal of Artificial Intelligence Research*, 69:1287–1332, 2020.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In Cohen, W. W. and Hirsh, H. (eds.), *Machine Learning Proceedings 1994*, pp. 157–163. Morgan Kaufmann, San Francisco (CA), 1994. ISBN 978-1-55860-335-6. doi: https://doi.org/10.1016/B978-1-55860-335-6.50027-1. URL https://www.sciencedirect.com/science/article/pii/B9781558603356500271.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention, 2020.

Lonini, L., Forestier, S., Teulière, C., Zhao, Y., Shi, B. E., and Triesch, J. Robust active binocular vision through intrinsically motivated learning. *Frontiers in neurorobotics*, 7:20, 2013.

Lopes, M., Lang, T., Toussaint, M., and Oudeyer, P. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Proc. of NeurIPS*, pp. 206–214, 2012.

Lowe, R., WU, Y., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Proc. of NeurIPS*, 30: 6379–6390, 2017.

Luketina, J., Nardelli, N., Farquhar, G., Foerster, J. N., Andreas, J., Grefenstette, E., Whiteson, S., and Rocktäschel, T. A survey of reinforcement learning informed by natural language. In *Proc. of IJCAI*, pp. 6309–6317, 2019.

Lupyan, G. Carving Nature at Its Joints and Carving Joints into Nature: How Labels Augment Category Representations. In *Modeling Language, Cognition and Action*. World Scientific, 2005. ISBN 978-981-256-324-8 978-981-270-188-6.

Lupyan, G. What Do Words Do? Toward a Theory of Language-Augmented Thought. In *Psychology of Learning and Motivation*. Elsevier, 2012.

Lynch, C. and Sermanet, P. Grounding language in play. ArXiv - abs/2005.07648, 2020.

Lynch, C. and Sermanet, P. Language conditioned imitation learning over unstructured data. *Robotics: Science and Systems XVII*, 2021.

Lynch, C., Khansari, M., Xiao, T., Kumar, V., Tompson, J., Levine, S., and Sermanet, P. Learning latent plans from play. In *Proceedings of the Conference on Robot Learning*, volume 100, pp. 1113–1132, 2020.

Madden, C., Hoen, M., and Dominey, P. F. A cognitive neuroscience perspective on embodied language for human–robot cooperation. *Brain and Language*, 112(3):180–188, mar 2010. ISSN 0093-934X. doi: 10.1016/J.BANDL.2009.07.001.

Mangin, O., Filliat, D., Ten Bosch, L., and Oudeyer, P.-Y. Mca-nmf: Multimodal concept acquisition with non-negative matrix factorization. *PloS one*, 10(10):e0140732, 2015.

Mankowitz, D. J., Žídek, A., Barreto, A., Horgan, D., Hessel, M., Quan, J., Oh, J., van Hasselt, H., Silver, D., and Schaul, T. Unicorn: Continual learning with a universal, off-policy agent. ArXiv - abs/1802.08294, 2018.

Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences from Natural Supervision. *ArXiv – abs/1904.12584*, 2019.

Martius, G., Der, R., and Ay, N. Information driven self-organization of complex robotic behaviors. *PloS one*, 8(5):e63400, 2013.

Marzoev, A., Madden, S., Kaashoek, M. F., Cafarella, M., and Andreas, J. Unnatural language processing: Bridging the gap between synthetic and natural language data, 2020.

Masquil, E., Hamon, G., Nisioti, E., and Moulin-Frier, C. Intrinsically-motivated goal-conditioned reinforcement learning in multi-agent environments, 2022.

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., and Smith, L. B. Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends in cognitive sciences*, 14(8):348–356, 2010.

McClelland, J. L., Hill, F., Rudolph, M., Baldridge, J., and Schütze, H. Placing language in an integrated understanding system: Next steps toward human-level performance in neural language models. *Proceedings of the National Academy of Sciences*, 117(42): 25966–25974, 2020.

Mcclung, J., Placì, S., Bangerter, A., Clément, F., and Bshary, R. The language of cooperation: Shared intentionality drives variation in helping as a function of group membership. *Proceedings of the Royal Society B: Biological Sciences*, 284:20171682, 09 2017. doi: 10.1098/rspb.2017.1682.

McDowell, J. *Mind and World*. Harvard University Press, 1996.

Mihai, D. and Hare, J. S. Differentiable drawing and sketching. *ArXiv*, abs/2103.16194, 2021a.

Mihai, D. and Hare, J. S. Learning to draw: Emergent communication through sketching. *NeurIPS*, 2021b.

Mintz, T. H. Frequent frames as a cue for grammatical categories in child directed speech. *Cognition*, 90(1):91–117, 2003.

Mirchandani, S., Karamcheti, S., and Sadigh, D. ELLA: Exploration through Learned Language Abstraction. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 29529–29540. Curran Associates, Inc., 2021.

Mirolli, M. and Parisi, D. Towards a Vygotskyan Cognitive Robotics: The Role of Language as a Cognitive Tool. *New Ideas in Psychology*, 29(3), 2011. ISSN 0732118X.

Mishra, J. and Gazzaley, A. Closed-loop cognition: the next frontier arrives. *Trends in Cognitive Sciences*, 19:242–243, 2015.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Mohamed, S. and Rezende, D. J. Variational information maximisation for intrinsically motivated reinforcement learning. In *Proc. of NeurIPS*, pp. 2125–2133, 2015.

Montague, R. Universal grammar. *Theoria*, 36(3), 1970.

Mordatch, I. and Abbeel, P. Emergence of grounded compositional language in multi-agent populations. In *AAAI*, 2018.

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. Experimental evidence for the co-evolution of hominin tool-making teaching and language. *Nature communications*, 6(1):1–8, 2015.

Moulin-Frier, C. *The Ecology of Open-Ended Skill Acquisition*. PhD thesis, Université de Bordeaux (UB), 2022. URL https://hal.inria.fr/tel-03875448.

Moulin-Frier, C. and Oudeyer, P.-Y. Multi-agent reinforcement learning as a computational tool for language evolution research: Historical context and future challenges. *ArXiv*, abs/2002.08878, 2020.

Moulin-Frier, C., Nguyen, S. M., and Oudeyer, P.-Y. Self-organization of early vocal development in infants and machines: The role of intrinsic motivation. *Frontiers in Psychology (Cognitive Science)*, 4(1006), 2014. ISSN 1664-1078.

Moulin-Frier, C., Diard, J., Schwartz, J.-L., and Bessière, P. Cosmo ("communicating about objects using sensory–motor operations"): A bayesian modeling framework for studying speech communication and the emergence of phonological systems. *Journal of Phonetics*, 53:5–41, 2015. ISSN 0095-4470. doi: https://doi.org/10.1016/j.wocn.2015.06.001. URL https://www.sciencedirect.com/science/article/pii/S0095447015000352. On the cognitive nature of speech sound systems.

Mu, J. and Goodman, N. Emergent communication of generalizations. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17994–18007. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/9597353e41e6957b5e7aa79214fcb256-Paper.pdf.

Mu, J., Zhong, V., Raileanu, R., Jiang, M., Goodman, N. D., Rocktaschel, T., and Grefenstette, E. Improving Intrinsic Exploration with Language Abstractions. *ArXiv – abs/2202.08938*, 2022.

Muñoz-Moldes, S. and Cleeremans, A. Delineating implicit and explicit processes in neurofeedback learning. *Neuroscience & Biobehavioral Reviews*, 118:681–688, 2020. ISSN 0149-7634. doi: https://doi.org/10.1016/j.neubiorev.2020.09.003. URL https://www.sciencedirect.com/science/article/pii/S0149763420305595.

Nachum, O., Gu, S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In *Proc. of NeurIPS*, pp. 3307–3317, 2018.

Nair, A., McGrew, B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6292–6299. IEEE, 2018a.

Nair, A., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S. Visual Reinforcement Learning with Imagined Goals. *Proc. of NeurIPS*, 2018b.

Nair, A., Bahl, S., Khazatsky, A., Pong, V., Berseth, G., and Levine, S. Contextual imagined goals for self-supervised robotic learning. In *Conference on Robot Learning*, pp. 530–539, 2020.

Narayan-Chen, A., Jayannavar, P., and Hockenmaier, J. Collaborative dialogue in minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5405–5415, 2019.

Ndousse, K. K., Eck, D., Levine, S., and Jaques, N. Emergent social learning via multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2021.

Neu, G. and Szepesvári, C. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, UAI'07, pp. 295–302, Arlington, Virginia, USA, 2007. AUAI Press. ISBN 0974903930.

Ng, A. Y. and Russell, S. J. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

Nguyen, K., Misra, D., Schapire, R., Dudík, M., and Shafto, P. Interactive learning from activity description. *Proc. of ICML*, 2021.

Nguyen, M. and Oudeyer, P.-Y. Socially guided intrinsic motivation for robot learning of motor skills. *Autonomous Robots*, 36(3):273–294, 2014.

Niu, Y., Paleja, R., and Gombolay, M. Multi-agent graph-attention communication and teaming. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, pp. 964–973, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.

Oh, J., Singh, S. P., Lee, H., and Kohli, P. Zero-shot task generalization with multi-task deep reinforcement learning. In *Proc. of ICML*, volume 70, pp. 2661–2670, 2017.

Oh, J., Guo, Y., Singh, S., and Lee, H. Self-imitation learning. In *ICML*, 2018.

Oliphant, M. and Batali, J. Learning and the emergence of coordinated communication. *Center for Research on Language Newsletter*, 11, 03 1997.

Oller, D. K., Griebel, U., Iyer, S. N., Jhang, Y., Warlaumont, A. S., Dale, R., and Call, J. Language origins viewed in spontaneous and interactive vocal rates of human and bonobo infants. *Frontiers in psychology*, 10:729, 2019.

OroojlooyJadid, A. and Hajinezhad, D. A review of cooperative multi-agent deep reinforcement learning, 2019. URL https://arxiv.org/abs/1908.03963.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., Peters, J., et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

Oudeyer, P.-Y. The self-organization of speech sounds. *Journal of theoretical biology*, 233 3:435–49, 2005.

Oudeyer, P.-Y. Self-organization in the evolution of speech. In *Oxford Studies in the Evolution of Language*, 2006.

Oudeyer, P.-Y. and Kaplan, F. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2007.

Oudeyer, P.-Y. and Smith, L. B. How evolution may work through curiosity-driven developmental process. *Topics in Cognitive Science*, 8(2):492–502, 2016.

Pagin, P. and Westerståhl, D. Compositionality i: Definitions and variants. *Philosophy Compass*, 5(3):250–264, 2010. doi: https://doi.org/10.1111/j.1747-9991.2009.00228.x. URL https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/j.1747-9991.2009.00228.x.

Pashevich, A., Schmid, C., and Sun, C. Episodic Transformer for Vision-and-Language Navigation. *ArXiv – abs/2105.06453*, 2021.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *Proc. of ICML*, volume 70, pp. 2778–2787, 2017.

Paul, R., Arkin, J., Roy, N., and Howard, T. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. In *Robotics: Science and Systems*, 2016.

Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A. Film: Visual reasoning with a general conditioning layer, 2017.

Pérolat, J., Leibo, J. Z., Zambaldi, V., Beattie, C., Tuyls, K., and Graepel, T. A multi-agent reinforcement learning model of common-pool resource appropriation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/2b0f658cbffd284984fb11d90254081f-Paper.pdf.

Pfeifer, R., Lungarella, M., and Iida, F. Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093, 2007. doi: 10.1126/science.1145803. URL https://www.science.org/doi/abs/10.1126/science.1145803.

Piaget, J. *The Origins of Intelligence in Children*. Translation Margaret Cook – WW Norton & Co, 1952.

Pinker, S. Pinker s., language learnability and language development. cambridge ma: Harvard university press, 1984. pp. xi 435. *Journal of Child Language*, 15(1):189–199, 1988. doi: 10.1017/S0305000900012137.

Pitis, S., Chan, H., Zhao, S., Stadie, B. C., and Ba, J. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *Proc. of ICML*, volume 119, pp. 7750–7761, 2020.

Plappert, M., Andrychowicz, M., Ray, A., McGrew, B., Baker, B., Powell, G., Schneider, J., Tobin, J., Chociej, M., Welinder, P., et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. ArXiv - abs/1802.09464, 2018.

Polyak, B. T. and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. doi: 10.1137/0330046.

Pomerleau, D. A. Alvinn: An autonomous land vehicle in a neural network. In Touretzky, D. (ed.), *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988. URL https://proceedings.neurips.cc/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf.

Pomerleau, D. A. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

Pong, V., Dalal, M., Lin, S., Nair, A., Bahl, S., and Levine, S. Skew-fit: State-covering self-supervised reinforcement learning. In *Proc. of ICML*, volume 119, pp. 7783–7792, 2020.

Portelance, E., Frank, M. C., Jurafsky, D., Sordoni, A., and Laroche, R. The emergence of the shape bias results from communicative efficiency. In *CONLL*, 2021.

Portelas, R., Colas, C., Hofmann, K., and Oudeyer, P.-Y. Teacher Algorithms for Curriculum Learning of Deep Rl in Continuously Parameterized Environments. In *Proc. of CoRL*, pp. 835–853, 2020a.

Portelas, R., Colas, C., Weng, L., Hofmann, K., and Oudeyer, P. Automatic curriculum learning for deep RL: A short survey. In *Proc. of IJCAI*, pp. 4819–4825, 2020b.

Precup, D. *Temporal abstraction in reinforcement learning*. PhD thesis, The University of Massachussetts, 2000.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Racanière, S., Lampinen, A., Santoro, A., Reichert, D., Firoiu, V., and Lillicrap, T. Automated curricula through setter-solver interactions. ArXiv - abs/1909.12892, 2019.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning Transferable Visual Models from Natural Language Supervision. *Proc. of ICML*, 2021.

Raileanu, R. and Rocktäschel, T. RIDE: rewarding impact-driven exploration for procedurally-generated environments. In *Proc. of ICLR*, 2020.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-Shot Text-to-Image Generation. *ArXiv – abs/2102.12092*, 2021. doi: 10.48550/ARXIV.2102.12092. URL https://arxiv.org/abs/2102.12092.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *ArXiv – abs/2204.06125*, 2022.

Ratliff, N. D., Bagnell, J. A., and Zinkevich, M. A. Maximum margin planning. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 729–736, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi: 10.1145/1143844.1143936. URL https://doi.org/10.1145/1143844.1143936.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-maron, G., Giménez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. A generalist agent. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=1ikK0kHjvj. Featured Certification.

Ren, Y., Guo, S., Labeau, M., Cohen, S. B., and Kirby, S. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkePNpVKPB.

Röder, F., Eppe, M., Nguyen, P. D., and Wermter, S. Curious hierarchical actor-critic reinforcement learning. In *International Conference on Artificial Neural Networks*, pp. 408–419. Springer, 2020.

Rodríguez Luna, D., Ponti, E. M., Hupkes, D., and Bruni, E. Internal and external pressures on language emergence: least effort, object constancy and frequency. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4428–4437, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.397. URL https://aclanthology.org/2020.findings-emnlp.397.

Rohlfing, K. J., Wrede, B., Vollmer, A.-L., and Oudeyer, P.-Y. An Alternative to Mapping a Word onto a Concept in Language Acquisition: Pragmatic Frames. *Frontiers in Psychology*, 7, 2016. ISSN 1664-1078.

Rolf, M. and Steil, J. J. Efficient exploratory learning of inverse kinematics on a bionic elephant trunk. *IEEE transactions on neural networks and learning systems*, 25(6): 1147–1160, 2013.

Rolf, M., Steil, J. J., and Gienger, M. Goal babbling permits direct learning of inverse kinematics. *IEEE Transactions on Autonomous Mental Development*, 2(3):216–229, 2010.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *ArXiv*, abs/2112.10752, 2021.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In Gordon, G., Dunson, D., and Dudík, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

Roy, J., Barde, P., Harvey, F., Nowrouzezahrai, D., and Pal, C. Promoting coordination through policy regularization in multi-agent deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15774–15785, 2020.

Ruis, L., Andreas, J., Baroni, M., Bouchacourt, D., and Lake, B. M. A benchmark for systematic generalization in grounded language understanding. In *Proc. of NeurIPS*, 2020.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., and Hinton, G. Sequential Thought Processes in Pdp Models. *Parallel distributed processing: explorations in the microstructures of cognition*, 2:3–57, 1986.

Runco, M. A. and Jaeger, G. J. The Standard Definition of Creativity. *Creativity Research Journal*, (1), 2012. ISSN 1040-0419, 1532-6934.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022.

Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. A simple neural network module for relational reasoning, 2017.

Santoro, A., Lampinen, A., Mathewson, K., Lillicrap, T., and Raposo, D. Symbolic behaviour in artificial intelligence. *ArXiv – abs/2102.03406*, 2021.

Santucci, V. G., Baldassarre, G., and Mirolli, M. Grail: a goal-discovering robotic architecture for intrinsically-motivated learning. *IEEE Transactions on Cognitive and Developmental Systems*, 8(3):214–231, 2016.

Santucci, V. G., Oudeyer, P.-Y., Barto, A., and Baldassarre, G. Intrinsically motivated open-ended learning in autonomous robots. *Frontiers in Neurorobotics*, 13:115, 2020.

Schaal, S. Learning from demonstration. In Mozer, M., Jordan, M., and Petsche, T. (eds.), *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996. URL https://proceedings.neurips.cc/paper/1996/file/68d13cf26c4b4f4f932e3eff990093ba-Paper.pdf.

Schaal, S. Dynamic movement primitives -a framework for motor control in humans and humanoid robotics. 2006.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *Proc. of ICML*, volume 37, pp. 1312–1320, 2015.

Schmidhuber, J. 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*, pp. 222–227, 1991.

Schramowski, P., Turan, C., Andersen, N., Rothkopf, C. A., and Kersting, K. Large Pre-Trained Language Models Contain Human-Like Biases of What Is Right and Wrong to Do. *Nature Machine Intelligence*, (3), 2022.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models. In *Proc. of ICML*, volume 119, pp. 8583–8592, 2020.

Shah, D. S., Schwartz, H. A., and Hovy, D. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5248–5264, Online, 2020. Association for Computational Linguistics.

Shanahan, M. and Mitchell, M. Abstraction for Deep Reinforcement Learning. *Proc. of IJCAI*, 2022.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. In *Proc. of ICLR*, 2020.

Sharma, P., Torralba, A., and Andreas, J. Skill Induction and Planning with Latent Language. *Proc. of ACL*, 2021.

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., and Fox, D. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020.

Shridhar, M., Yuan, X., Cote, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. *Proc. of ICLR*, 2021.

Sigaud, O., Colas, C., Akakzia, A., Chetouani, M., and Oudeyer, P.-Y. Towards Teachable Autonomous Agents. *ArXiv – abs/2105.11977*, 2021.

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. Mastering the game of go with deep neural networks and tree search. In *nature*, volume 529, pp. 484–489. Nature Publishing Group, 2016.

Simpson, G. G. The baldwin effect. *Evolution*, 7(2):110–117, 1953. ISSN 00143820, 15585646. URL http://www.jstor.org/stable/2405746.

Smolensky, P. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1):159–216, 1990. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(90)90007-M. URL https://www.sciencedirect.com/science/article/pii/000437029090007M.

Spelke, E. S., Breinlinger, K., Macomber, J., and Jacobson, K. Origins of knowledge. *Psychological review*, 99(4):605, 1992.

Sperber, D., Premack, D., and Premack, A. J. *Causal Cognition: A Multidisciplinary Debate*. Clarendon Press Oxford, 1995.

Steels, L. A self-organizing spatial vocabulary. *Artificial life*, 2(3):319–332, 1995a.

Steels, L. The synthetic modeling of language origins. *Evolution of Communication Journal*, 1, 10 1997. doi: 10.1075/eoc.1.1.02ste.

Steels, L. Semiotic dynamics for embodied agents. *IEEE Intelligent Systems*, 21(3): 32–38, 2006. doi: 10.1109/MIS.2006.58.

Steels, L. L. A self-organizing spatial vocabulary. *Artificial Life*, 2:319–332, 1995b.

Steels, L. L. Language games for autonomous robots. *IEEE Intelligent Systems*, 16: 16–22, 2001.

Steels, L. L. *The Talking Heads experiment*. Number 1 in Computational Models of Language Evolution. Language Science Press, Berlin, 2015. doi: 10.17169/FUDOCS_document_000000022455.

Steels, L. L. and Loetzsch, M. The grounded naming game. 2012.

Stevens, K. N. On the quantal nature of speech. *Journal of Phonetics*, 17(1):3–45, 1989. ISSN 0095-4470. doi: https://doi.org/10.1016/S0095-4470(19)31520-7. URL https://www.sciencedirect.com/science/article/pii/S0095447019315207.

Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., et al. Open-ended learning leads to generally capable agents. ArXiv - abs/2107.12808, 2021.

Sugita, Y. and Tani, J. Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive behavior*, 13(1):33–52, 2005.

Sukhbaatar, S., Szlam, A. D., and Fergus, R. Learning multiagent communication with backpropagation. In *NIPS*, 2016.

Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., and Fergus, R. Intrinsic motivation and automatic curricula via asymmetric self-play. In *Proc. of ICLR*, 2018.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Sutton, R. S., Precup, D., and Singh, S. P. Intra-option learning about temporally abstract actions. In *Proc. of ICML*, volume 98, pp. 556–564, 1998.

Sutton, R. S., Precup, D., and Singh, S. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial intelligence*, (1-2), 1999. Publisher: Elsevier.

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. 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*, pp. 761–768, 2011.

Szabó, Z. G. Compositionality. In Zalta, E. N. and Nodelman, U. (eds.), *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2022 edition, 2022.

Számadó, S. Pre-hunt communication provides context for the evolution of early human language. *Biological Theory*, 5:366–382, 01 2010. doi: 10.1162/BIOT_a_00064.

Tam, A. C., Rabinowitz, N. C., Lampinen, A. K., Roy, N. A., Chan, S. C. Y., Strouse, D., Wang, J. X., Banino, A., and Hill, F. Semantic Exploration from Language Abstractions and Pretrained Representations. *ArXiv – abs/2204.05080*, 2022.

Tani, J. *Exploring robotic minds: actions, symbols, and consciousness as self-organizing dynamic phenomena*. Oxford University Press, 2016.

Taniguchi, T., Nagai, T., Nakamura, T., Iwahashi, N., Ogata, T., and Asoh, H. Symbol emergence in robotics: a survey. *Advanced Robotics*, 30(11-12):706–728, 2016.

Tomasello, M. *The cultural origins of human cognition*. Harvard University Press, 1999a. ISBN 9780674005822.

Tomasello, M. *The Cultural Origins of Human Cognition*. Harvard University Press, 1999b. ISBN 978-0-674-00582-2.

Tomasello, M. The item-based nature of children's early syntactic development. *Trends in cognitive sciences*, 4(4):156–163, 2000.

Tomasello, M. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, 2005. ISBN 9780674017641.

Tomasello, M. *Becoming Human – A Theory of Ontogeny*. Harvard University Press, Cambridge, MA and London, England, 2019. ISBN 9780674988651. doi: doi:10.4159/9780674988651. URL https://doi.org/10.4159/9780674988651.

Tomasello, M. and Olguin, R. Twenty-three-month-old children have a grammatical category of noun. *Cognitive development*, 8(4):451–464, 1993.

Tomasello, M., Carpenter, M., Call, J., Behne, T., and Moll, H. Understanding and Sharing Intentions: The Origins of Cultural Cognition. *Behavioral and brain sciences*, 2005.

Tuci, E., Ferrauto, T., Zeschel, A., Massera, G., and Nolfi, S. An experiment on behavior generalization and the emergence of linguistic compositionality in evolving robots. *IEEE Transactions on Autonomous Mental Development*, 3(2):176–189, 2011.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proc. of NeurIPS*, pp. 5998–6008, 2017.

Veeriah, V., Oh, J., and Singh, S. Many-goals reinforcement learning. ArXiv - abs/1806.09605, 2018.

Venkattaramanujam, S., Crawford, E., Doan, T., and Precup, D. Self-supervised learning of distance functions for goal-conditioned reinforcement learning. 2019.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *Proc. of ICML*, volume 70, pp. 3540–3549, 2017.

Vollmer, A.-L., Grizou, J., Lopes, M., Rohlfing, K., and Oudeyer, P.-Y. Studying the co-construction of interaction protocols in collaborative tasks with humans. In *4th International Conference on Development and Learning and on Epigenetic Robotics*, pp. 208–215. IEEE, 2014.

Vollmer, A.-L., Wrede, B., Rohlfing, K. J., and Oudeyer, P.-Y. Pragmatic frames for teaching and learning in human–robot interaction: Review and challenges. *Frontiers in neurorobotics*, 10:10, 2016.

von Foerster, H. *On Self-Organizing Systems and Their Environments*, pp. 1–19. Springer New York, New York, NY, 2003. ISBN 978-0-387-21722-2. doi: 10.1007/0-387-21722-3_1. URL https://doi.org/10.1007/0-387-21722-3_1.

Vygotsky, L. S. Play and Its Role in the Mental Development of the Child. *Soviet Psychology*, 1933.

Vygotsky, L. S. *Thought and Language*. MIT press, 1934.

Vygotsky, L. S. Tool and Symbol in Child Development. In *Mind in Society*, chapter Tool and Symbol in Child Development, pp. 19–30. Harvard University Press, 1978. ISBN 0674576292.

Vyshedskiy, A. Language Evolution to Revolution: the Leap From Rich-Vocabulary Non-Recursive Communication System to Recursive Language 70,000 Years Ago Was Associated with Acquisition of a Novel Component of Imagination, Called Prefrontal Synthesis, Enabled By a Mutation that Slowed Down the Prefrontal Cortex Maturation Simultaneously in Two or More Children – the Romulus and Remus Hypothesis. *Research Ideas and Outcomes*, 2019. ISSN 2367-7163.

Warde-Farley, D., de Wiele, T. V., Kulkarni, T. D., Ionescu, C., Hansen, S., and Mnih, V. Unsupervised Control Through Non-Parametric Discriminative Rewards. In *Proc. of ICLR*, 2019.

Watkins, C. J. C. H. and Dayan, P. Q-learning. 8(3):279–292, 1992. ISSN 1573-0565. doi: 10.1007/BF00992698. URL https://doi.org/10.1007/BF00992698.

Waxman, S. R. and Markow, D. B. Words as Invitations to Form Categories: Evidence from 12-to 13-Month-Old Infants. *Cognitive psychology*, (3), 1995.

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., Kenton, Z., Brown, S., Hawkins, W., Stepleton, T., Biles, C., Birhane, A., Haas, J., Rimell, L., Hendricks, L. A., Isaac, W., Legassick, S., Irving, G., and Gabriel, I. Ethical and social risks of harm from language models. *ArXiv – abs/2112.04359*, 2021.

Wellman, H. M. *The child's theory of mind.* The MIT Press, 1992.

West, P., Bhagavatula, C., Hessel, J., Hwang, J. D., Jiang, L., Bras, R. L., Lu, X., Welleck, S., and Choi, Y. Symbolic Knowledge Distillation: from General Language Models to Commonsense Models. *Proc. of NAACL*, 2022.

Whorf, B. L. *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf.* MIT press, 1956.

Wiessner, P. W. Embers of society: Firelight talk among the ju/'hoansi bushmen. *Proceedings of the National Academy of Sciences*, 111(39):14027–14035, 2014.

Wittgenstein, L. *Philosophical Investigations.* John Wiley & Sons, 1953.

Witty, S., Lee, J. K., Tosch, E., Atrey, A., Clary, K., Littman, M. L., and Jensen, D. Measuring and Characterizing Generalization in Deep Reinforcement Learning. *Applied AI Letters*, 2021.

Wong, C., Ellis, K., Tenenbaum, J. B., and Andreas, J. Leveraging Language to Learn Program Abstractions and Search Heuristics. *Proc. of ICML*, 2021.

Woodward, M., Finn, C., and Hausman, K. Learning to interactively learn and assist. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 2535–2543, 2020.

Xie, T., Langford, J., Mineiro, P., and Momennejad, I. Interaction-grounded learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11414–11423. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/xie21e.html.

Yan, C., Carnevale, F., Georgiev, P., Santoro, A., Guy, A., Muldal, A., Hung, C.-C., Abramson, J., Lillicrap, T., and Wayne, G. Intra-agent speech permits zero-shot task acquisition. *ArXiv – abs/2206.03139*, 2022.

Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., and Tenenbaum, J. B. Clevrer: Collision events for video representation and reasoning, 2020.

Yoshida, H. and Smith, L. B. Sound Symbolism and Early Word Learning in Two Languages. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2003.

Yuan, X., Côté, M.-A., Fu, J., Lin, Z., Pal, C., Bengio, Y., and Trischler, A. Interactive Language Learning by Question Answering. In *Proc. of EMNLP*. Association for Computational Linguistics, 2019.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R., and Smola, A. J. Deep sets. In *Proc. of NeurIPS*, pp. 3391–3401, 2017.

Zaremba, W., Sutskever, I., and Vinyals, O. Recurrent neural network regularization. *ArXiv - abs/1409.2329*, 2014.

Zeng, A., Wong, A., Welker, S., Choromanski, K., Tombari, F., Purohit, A., Ryoo, M., Sindhwani, V., Lee, J., Vanhoucke, V., et al. Socratic models: Composing zero-shot multimodal reasoning with language. *ArXiv – abs/2204.00598*, 2022.

Zhang, Y., Abbeel, P., and Pinto, L. Automatic curriculum learning through value disagreement. In *Proc. of NeurIPS*, 2020.

Zhou, H., Kadav, A., Lai, F., Niculescu-Mizil, A., Min, M. R., Kapadia, M., and Graf, H. P. Hopper: Multi-hop transformer for spatiotemporal reasoning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=MaZFq7bJif7.

Zhou, L. and Small, K. Inverse reinforcement learning with natural language goals, 2020a.

Zhou, L. and Small, K. Inverse Reinforcement Learning with Natural Language Goals. *ArXiv – abs/2008.06924*, 2020b.

Zhu, C., Dastani, M., and Wang, S. A survey of multi-agent reinforcement learning with communication, 2022.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3357–3364. IEEE, 2017.

Ziebart, B., Maas, A., Bagnell, J., and Dey, A. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2008.

Zlatev, J. The Epigenesis of Meaning in Human Beings, and Possibly in Robots. *Minds and Machines*, (2), 2001. ISSN 1572-8641.

Zuidema, W. and De Boer, B. The evolution of combinatorial phonology. *Journal of Phonetics*, 37(2):125–144, 2009.

Zwaan, R. and Madden, C. Embodied sentence comprehension. *Grounding Cognition: The Role of Perception and Action in Memory, Language, and Thinking*, pp. 224–245, 2005a. doi: 10.1017/CBO9780511499968.010.

Zwaan, R. A. and Madden, C. J. *Embodied Sentence Comprehension*, pp. 224–245. Cambridge University Press, 2005b. doi: 10.1017/CBO9780511499968.010.