Modeling the neural network responsible for song learning

Silvia Pagliarini

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

DOCTEUR DE

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE

SPÉCIALITÉ
Informatique

Par Silvia PAGLIARINI

MODELING THE NEURAL NETWORK RESPONSIBLE FOR SONG LEARNING

Sous la direction de Xavier HINAUT
et de Arthur LEBLOIS

Soutenue le 25, Mars, 2021

Membres du jury :

Mme. DESAINTE-CATHERINE, Myriam	University of Bordeaux	Presidente
M. HAHNLOSER, Richard	ETH Zurich	Rapporteur
M. SCHWARTZ, Jean-Luc	CNRS	Rapporteur
Mme. WARLAUMONT, Anne	University of California, Los Angeles	Examinatrice

Résumé vulgarisé

L'apprentissage de la parole chez les enfants est similaire à celle du chant chez les oiseaux. Ils passent par les mêmes phases de développement: ils commencent par écouter leurs parents, puis ils essayent de reproduire leurs vocalisations en babillant, pour enfin produire des sons de plus en plus proches de ceux de leurs parents. L'oiseau chanteur a des circuits cérébraux spécifiques dédiés à cet apprentissage, ce qui en fait un modèle idéal pour explorer les mécanismes neuronaux de l'apprentissage vocal par imitation. Dans cette thèse, on modélise cet apprentissage à l'aide d'une boucle perceptuo-motrice dans laquelle un mécanisme d'évaluation sensorielle guide l'apprentissage. On utilise notamment des développements récents de l'intelligence artificielle (réseaux antagonistes génératifs) pour produire les chants d'oiseaux. Cela nous permet de mieux comprendre l'apprentissage vocal par imitation, et plus généralement l'apprentissage sensorimoteur.

English version

Humans learn to speak in a similar way as how songbirds learn to sing. Both learn to speak/sing by imitation from an early age going through the same stages of development. First they listen to their parents' vocalizations, then they try to reproduce them: initially babbling, until their vocal output mimics those of their parents. Songbirds have dedicated brain circuits for vocal learning, making them an ideal model for exploring the representation of imitative vocal learning. My research project aims to build a bio-inspired model to describe imitative vocal learning. This model consists in a perceptual-motor loop where a sensory evaluation mechanism drives learning. The sound production is obtained from real recordings, using recent developments in artificial intelligence. This project, in between computer science and neuroscience, may help to better understand imitative vocal learning, and more generally sensorimotor learning.

Titre : Modélisation du réseau neuronal responsable de l'apprentissage du chant chez l'oiseau chanteur

Resumé

Pendant la première période de leur vie, les bébés et les jeunes oiseaux présentent des phases de développement vocal comparables : ils écoutent d'abord leurs parents/tuteurs afin de construire une représentation neurale du stimulus auditif perçu, puis ils commencent à produire des sons qui se rapprochent progressivement du chant de leur tuteur. Cette phase d'apprentissage est appelée la phase sensorimotrice et se caractérise par la présence de babillage. Elle se termine lorsque le chant se cristallise, c'est-à-dire lorsqu'il devient semblable à celui produit par les adultes.

Il y a des similitudes entre les voies cérébrales responsables de l'apprentissage sensorimoteur chez l'homme et chez les oiseaux. Dans les deux cas, une voie s'occupe de la production vocale et implique des projections directes des zones auditives vers les zones motrices, et une autre voie s'occupe de l'apprentissage vocal, de l'imitation et de la plasticité. Chez les oiseaux, ces circuits cérébraux sont exclusivement dédiés à l'apprentissage du chant, ce qui en fait un modèle idéal pour explorer les mécanismes neuronaux de l'apprentissage vocal par imitation.

Cette thèse vise à construire un modèle de l'apprentissage du chant des oiseaux par imitation. De nombreuses études antérieures ont tenté de mettre en œuvre l'apprentissage par imitation dans des modèles informatiques et partagent une structure commune. Ces modèles comprennent des mécanismes d'apprentissage et, éventuellement, des stratégies d'exploration et d'évaluation. Dans ces modèles, une fonction de contrôle moteur permet la production de sons et une réponse sensorielle modélise soit la façon dont le son est perçu, soit la façon dont il façonne la récompense. Les entrées et les sorties de ces fonctions sont dans plusieurs espaces: l'espace moteur (paramètres moteurs), l'espace sensoriel (sons réels), l'espace perceptif (représentation à faible dimension du son) ou l'espace des objectifs (représentation non perceptive du son cible).

Le premier modèle proposé est un modèle théorique inverse basé sur un modèle d'apprentissage vocal simplifié où l'espace sensoriel coïncide avec l'espace moteur (c'est-à-dire qu'il n'y a pas de production sonore). Une telle simplification permet d'étudier comment introduire des hypothèses biologiques (par exemple, une réponse non linéaire) dans un modèle d'apprentissage vocal et quels sont les paramètres qui influencent le plus la puissance de calcul du modèle. Afin de disposer d'un modèle complet (capable de percevoir et de produire des sons), nous avions besoin d'une fonction de contrôle moteur capable de reproduire des sons similaires à des données réelles. Nous avons analysé la capacité de WaveGAN (un réseau de génération) à produire des chants de canari réalistes. Dans ce modèle, l'espace d'entrée devient l'espace latent après l'entraînement et permet la représentation d'un ensemble de données à haute dimension dans une variété à plus basse dimension. Nous avons obtenu des chants de canari réalistes en utilisant seulement trois dimensions pour l'espace latent. Des analyses quantitatives et qualitatives démontrent les capacités d'interpolation du modèle, ce qui suggère que le modèle peut être utilisé comme fonction motrice dans un modèle d'apprentissage vocal. La deuxième version du

modèle est un modèle d'apprentissage vocal complet avec une boucle action-perception complète (il comprend l'espace moteur, l'espace sensoriel et l'espace perceptif). La production sonore est réalisée par le générateur GAN obtenu précédemment. Un réseau neuronal récurrent classant les syllabes sert de réponse sensorielle perceptive. La correspondance entre l'espace perceptuel et l'espace moteur est apprise par un modèle inverse. Les résultats préliminaires montrent l'impact du taux d'apprentissage lorsque différentes fonctions de réponse sensorielle sont mises en œuvre.

Mots clés :modélisation, modèle inverse, apprentissage sensori-moteur, oiseau chanteur, réseau de neurones artificiels, réseaux génératifs antagonistes

---

Title : Modeling the neural network responsible for song learning

Abstract

During the first period of their life, babies and juvenile birds show comparable phases of vocal development: first, they listen to their parents/tutors in order to build a neural representation of the experienced auditory stimulus, then they start to produce sound and progressively get closer to reproducing their tutor song. This phase of learning is called the sensorimotor phase and is characterized by the presence of babbling, in babies, and subsong, in birds. It ends when the song crystallizes and becomes similar to the one produced by the adults.

It is possible to find analogies between brain pathways responsible for sensorimotor learning in humans and birds: a vocal production pathway involves direct projections from auditory areas to motor neurons, and a vocal learning pathway is responsible for imitation and plasticity. The behavioral studies and the neuroanatomical structure of the vocal control circuit in humans and birds provide the basis for bio-inspired models of vocal learning. In particular, birds have brain circuits exclusively dedicated to song learning, making them an ideal model for exploring the representation of vocal learning by imitation of tutors.

This thesis aims to build a vocal learning model underlying song learning in birds. An extensive review of the existing literature is discussed in the thesis: many previous studies have attempted to implement imitative learning in computational models and share a common structure. These learning architectures include the learning mechanisms and, eventually, exploration and evaluation strategies. A motor control function enables sound production and sensory response models either how sound is perceived or how it shapes the reward. The inputs and outputs of these functions lie (1) in the motor space (motor parameters' space), (2) in the sensory space (real sounds) and (3) either in the perceptual space (a low dimensional representation of the sound) or in the internal representation of goals (a non-perceptual representation of the target sound).

The first model proposed in this thesis is a theoretical inverse model based on a simplified vocal learning model where the sensory space coincides with the motor space

(i.e., there is no sound production). Such a simplification allows us to investigate how to introduce biological assumptions (e.g. non-linearity response) into a vocal learning model and which parameters influence the computational power of the model the most. The influence of the sharpness of auditory selectivity and the motor dimension are discussed.

To have a complete model (which is able to perceive and produce sound), we needed a motor control function capable of reproducing sounds similar to real data (e.g. recordings of adult canaries). We analyzed the capability of WaveGAN (a Generative Adversarial Network) to provide a generator model able to produce realistic canary songs. In this generator model, the input space becomes the latent space after training and allows the representation of a high-dimensional dataset in a lower-dimensional manifold. We obtained realistic canary sounds using only three dimensions for the latent space. Among other results, quantitative and qualitative analyses demonstrate the interpolation abilities of the model, which suggests that the generator model we studied can be used as a motor function in a vocal learning model.

The second version of the sensorimotor model is a complete vocal learning model with a full action-perception loop (i.e., it includes motor space, sensory space, and perceptual space). The sound production is performed by the GAN generator previously obtained. A recurrent neural network classifying syllables serves as the perceptual sensory response. Similar to the first model, the mapping between the perceptual space and the motor space is learned via an inverse model. Preliminary results show the influence of the learning rate when different sensory response functions are implemented.

Keywords : modeling, inverse model, sensorimotor learning, songbird, artificial neural network, generative adversarial network

---

IMN (Institut des maladies neurodégénératives de Bordeaux) . Université de Bordeaux. 146 Rue Léo Saignat – 33000, Bordeaux.
INRIA Bordeaux Sud-Ouest. 200 Avenue de la Vieille Tour – 33405 Talence
UMR 5800 – LABRI. Université de Bordeaux. 351, Cours de la Libération – 33405 Talence

Résumé détaillé de la thèse en langue française

Les humains, comme les oiseaux chanteurs, apprennent par imitation dès leur plus jeune âge (par exemple, apprentissage de la parole chez les enfants et du chant chez les oiseaux chanteurs) : ils sont capables de reproduire un stimulus sensoriel vécu (par exemple un son) en trouvant la commande motrice appropriée pour le reproduire. Comment aborder la définition d'un modèle d'apprentissage sensorimoteur ? Quels sont les principaux éléments nécessaires pour construire une représentation minimale ? Le problème difficile de l'apprentissage sensorimoteur pour les systèmes naturels et artificiels motive notre recherche sur l'apprentissage vocal imitatif. Les études comportementales et la structure neuroanatomique du circuit de contrôle vocal chez les humains et les oiseaux servent de base pour un modèle bio-inspiré d'apprentissage vocal. En particulier, les oiseaux ont des circuits cérébraux exclusivement dédiés à l'apprentissage du chant, ce qui en fait un modèle idéal pour explorer la représentation de l'apprentissage vocal par imitation.

Cette thèse vise à construire un modèle bio-inspiré pour décrire l'apprentissage vocal par imitation chez les oiseaux en utilisant les récents développements de l'intelligence artificielle (par exemple les réseaux antagonistes génératifs, dénommés GAN). Un travail de revue de la littérature a été mené afin de comprendre en profondeur les concepts fondamentaux sur lesquels la thèse est basée. Le **Chapitre 1** contient une introduction à ces concepts: l'apprentissage sensorimoteur, l'apprentissage vocal d'un point de vue biologique et informatique et les modèles génératifs.

La littérature sur les modèles d'apprentissage vocal est vaste, variée et pleine de contenus multidisciplinaires. Le **Chapitre 2** compare les modèles d'apprentissage vocal existants issus d'études sur différents sujets (humains ou oiseaux chanteurs) et décrit la représentation d'un modèle d'apprentissage vocal minimal et comment il a été mis en œuvre dans la littérature. Tout d'abord, il contient une introduction à la neuroanatomie du cerveau humain et des oiseaux chanteurs, ainsi qu'une analyse des liens entre la biologie et les composants sensorimoteurs. Ensuite, il contient la description des composantes des modèles présents dans la littérature, et comment les modèles examinés peuvent être décomposés en fonction de ceux-ci. La comparaison entre plusieurs études a été effectuée avec soin et les différences/similitudes ont été largement discutées. Deux tableaux contiennent les détails de la comparaison. L'objectif est de trouver un scénario commun pour de démêler les compostants des modèles afin de mieux les comparer entre eux.

Un bref compte-rendu préliminaire a été publié précédemment dans le cadre d'un workshop de la conférence internationale ICDL-Epirob sur l'apprentissage sensorimoteur continu et non-supervisé (septembre 2018, Tokyo, Japon) et a ensuite été étendu au Chapitre 2. *Vocal Imitation in Sensorimotor Learning Models A Comparative Review* (Pagliarini et al., 2020) a été publié dans le Journal of Transactions on Cognitive and Developmental Systems, SI : Continual Unsupervised Sensorimotor Learning.

Le modèle d'apprentissage vocal le plus simple ne comprend que deux espaces : l'espace perceptuel (représentant par exemple les aires auditives du cerveau de l'oiseau) et l'espace moteur (représentant par exemple les aires motrices du cerveau de l'oiseau). Dans ce type de modèle, l'espace sensoriel et l'espace moteur se retrouvent être un seul et même espace.

Le premier modèle proposé dans cette thèse est un modèle inverse théorique basé sur un modèle d'apprentissage vocal simplifié où l'espace sensoriel coïncide avec l'espace moteur (c'est-à-dire qu'il n'y a pas de production sonore). Le **Chapitre 3** décrit l'architecture d'un modèle simple d'apprentissage vocal dans lequel une réponse sensorielle non linéaire active les unités perceptuelles et une règle d'apprentissage hébbienne normalisée pilote l'apprentissage. Ce modèle s'inspire du modèle théorique d'apprentissage vocal proposé précédemment par Hanuschkin et al. (2013) et Hahnloser and Ganguli (2013). L'architecture d'apprentissage est basée sur un appariement de la zone sensorielle à la zone motrice, apprise par une règle d'apprentissage hébbienne. L'exploration est aléatoire, et la réponse sensorielle permet l'activation des neurones sensoriels. Il est présenté un modèle théorique bio-inspiré de l'apprentissage vocal chez les oiseaux chanteurs et il reproduit numériquement les résultats théoriques. Dans ce chapitre, l'*aire perceptuel* correspond à la zone du cerveau où le stimulus perceptif est encodé et l'*aire motrice* correspond à la zone du cerveau d'où part l'entrée dans l'appareil moteur. En même temps, comme introduit dans le Chapitre 2, le stimulus appartient à l'espace perceptuel, est encodé dans l'espace perceptif et est produit grâce à une commande motrice appartenant à l'espace moteur.

Une telle simplification permet d'étudier comment introduire des hypothèses biologiques dans un modèle d'apprentissage vocal et quels sont les paramètres qui influencent le plus la puissance de calcul du modèle. La vitesse et la précision de l'apprentissage sont influencées par la sélectivité d'une part, et par la taille de l'aire motrice d'autre part. Cette dernière sera le point clé pour le développement ultérieur du modèle.

A Bio-inspired Model Towards Vocal Gesture Learning in Songbird" (Pagliarini et al., 2018a) a été publié dans les actes de l'ICDL-Epirob, et utilisé plusieurs fois pour des présentations orales et des sessions de posters. Plus de détails sont disponibles sur le site https://github.com/spagliarini/2018-ICDL-EPIROB.

Le modèle proposé dans le Chapitre 3 est construit en supposant qu'à chaque étape d'une nouvelle exploration motrice, une nouvelle réponse auditive est calculée et les poids synaptiques sont mis à jour en conséquence. En d'autres termes, les syllabes ont été considérées comme des entités qui durent un seul pas de temps, sans tenir compte du fait que, biologiquement, elles ont une certaine durée. En outre, le délai entre l'activité des motoneurones et celle des neurones auditifs (qui provoque le chevauchement entre la représentation auditive d'une syllabe et la production de la nouvelle syllabe) n'a pas été pris en compte. Une contribution pour décrire l'effet du retard du retour d'information auditif sur la précision de l'apprentissage se trouve dans une section dédiée qui contient des résultats complémentaires non publiés.

Pour disposer d'un modèle complet (c'est-à-dire capable de percevoir et de produire), il faut une fonction de contrôle moteur capable de reproduire des sons similaires à des données réelles (c'est-à-dire des enregistrements de canaris et de diamants mandarins, d'adultes et de jeunes).

Les modèles de variables latentes (ou modèles d'espace latent) sont une classe de modèles qui permettent de représenter des données en haute dimension en une représentation significative en basse dimension (appelée espace latent) où les points proches ont des propriétés similaires. En effet, l'espace latent est capable de représenter la variation des données réelles et d'encoder des éléments réalistes, y compris ceux qui ne font pas partie du jeu de données d'entraînement (Roberts et al., 2018). De plus, un espace latent pourrait permettre de générer un résultat réaliste en interpolant entre des points de l'espace latent (Radford et al., 2015). Toutes ces propriétés mettent en évidence la possibilité d'utiliser l'espace latent pour obtenir une représentation continue à faible dimension d'un ensemble de données donné.

Le **Chapitre 4** explore l'application d'un GAN à un ensemble de données de syllabes de canaris. WaveGAN (Donahue et al., 2018) a été entraîné sur un jeu de données vocales et sur des enregistrements dans la nature de plusieurs espèces d'oiseaux. Les résultats prometteurs obtenus sur un vaste ensemble de données très variables de chants d'oiseaux ont déterminé le choix d'étudier les performances du générateur WaveGAN sur un ensemble de données plus petit et plus propre, plus proche de l'ensemble de données vocales utilisé dans le travail original. D'une part, la capacité du générateur à reproduire des échantillons réalistes est confirmée par une analyse qualitative qui a été effectuée à la fin de l'entraînement. D'autre part, il met en évidence la possibilité d'utiliser un espace latent de faible dimension.

Comme mentionné dans le Chapitre 2, le modèle plus simple ne met pas en œuvre une fonction de contrôle moteur (c'est-à-dire que l'espace sensoriel coïncide avec l'espace moteur et qu'il n'y a pas de production sonore). Les modèles plus complexes définissent une fonction de contrôle moteur qui permet la production de sons. La fonction de réponse sensorielle traite le son et définit l'espace perceptuel. Enfin, l'architecture d'apprentissage décrit le lien entre l'espace perceptuel et l'espace moteur. L'objectif est de disposer d'un modèle basé sur une hypothèse biologique et réalisable par calcul (c'est-à-dire capable de converger vers l'apprentissage dans un laps de temps raisonnable et limité), contenant les éléments mentionnés ci-dessus et capable de produire un son réaliste en sortie.

Dans la littérature sur les oiseaux chanteurs, la fonction de contrôle moteur a souvent été définie à l'aide d'un système d'équations différentielles ordinaires qui modélisent l'anatomie du syrinx (c'est-à-dire l'organe vocal des oiseaux) (Amador et al., 2013) ou les caractéristiques du son (Doya, 2000). De tels modèles peuvent fournir des productions qualitativement bonnes mais ne sont pas toujours en mesure de reproduire parfaitement les connexions perceptuo-motrices (Pagliarini et al., 2020). Habituellement, les modèles mécanistes n'utilisent que quelques paramètres moteurs pour induire la plupart

des changements dans la production. Il est donc difficile de comprendre la correspondance entre les paramètres moteurs et la sortie. De plus, ils sont lents à simuler.

Le **Chapitre 5** propose un modèle complet d'apprentissage vocal. Le modèle comprend un espace moteur, un espace sensoriel et un espace perceptif, la fonction de contrôle moteur et la fonction de réponse sensorielle. Un modèle génératif de dimension 3 (c'est-à-dire que l'espace latent a 3 coordonnées) présenté dans le Chapitre 4 modélise la fonction de contrôle moteur, tandis que la combinaison d'un classificateur, basé sur un réseau neuronal récurrent (comme celui entraîné dans la section 4.3.4 du Chapitre 4) et une couche de normalisation, modélise la fonction de réponse sensorielle. Un point clé qu'il faut garder à l'esprit est la redondance de WaveGAN (Donahue et al., 2018) (et des GAN en général) : une syllabe peut être produite en utilisant de multiples configurations motrices. L'objectif du modèle est donc défini comme un objectif perceptuel. Les connexions entre l'espace moteur et l'espace perceptuel sont apprises par un modèle inverse. L'espace moteur est exploré à l'aide d'une exploration uniforme aléatoire et une simple règle d'apprentissage hébbienne dirige l'apprentissage. Le chapitre présente des résultats préliminaires sur l'influence du taux d'apprentissage lorsque différentes fonctions de réponse sensorielle sont mises en œuvre. Ces résultats ont été obtenus sur la base d'hypothèses simples et visent à être étendus. Des études plus approfondies de l'espace moteur (espace latent) sont nécessaires pour comprendre (1) comment il est structuré et quelle est sa topologie (où chaque classe de syllabes est située dans l'espace tridimensionnel), et (2) si la topologie particulière donnerait un indice pour une stratégie d'exploration particulière. Une simple règle d'apprentissage hébienne permet d'atteindre les objectifs perceptuels "cibles" mais ne comporte pas de critères d'arrêt. L'introduction d'un signal de renforcement pourrait aider l'apprentissage à se stabiliser après avoir atteint la région de l'espace moteur qui permet la production du but perceptuel correct.

Bien qu'il soit nécessaire de vérifier la capacité de chaque composante du modèle à traiter des données différentes, le modèle proposé au Chapitre 5 peut potentiellement aider à expliquer l'apprentissage vocal chez les oiseaux chanteurs, mais il est également ouvert à d'autres perspectives. Par exemple, la même structure de modèle pourrait être utilisée pour l'apprentissage vocal chez les humains ou dans la communication entre agents artificiels. Par exemple, WaveGAN, et en général les modèles génératifs, pourraient servir de fonction de contrôle moteur pour différents modèles d'apprentissage vocal. C'est-à-dire qu'on pourrait utiliser le même générateur (entraîné avec différents jeux de données) pour modéliser la fonction de contrôle moteur dans un modèle d'apprentissage vocal en essayant d'expliquer l'apprentissage des chansons dans les cas des canaries ou le développement de la parole chez les humains. Pour tester cette possibilité, le même modèle génératif devrait être entraîné sur différents ensembles de données afin d'évaluer sa capacité à reproduire des sorties réalistes.

Acknowledgements

I wish to express my sincere appreciation to my thesis supervisors, Arthur Leblois and Xavier Hinaut, who have guided me through this sometimes circuitous path of the doctoral program. First, for offering me to work with them on such an interesting topic. Secondly, for having accompanied me to the discovery of vocal learning with wisdom and patience, unraveling my doubts and insecurities. Without your continued commitment this journey would have been impossible to achieve. I am grateful for the curiosity and passion you have generated in me about a subject unknown to me until the day I set foot in Bordeaux.

I wish to show my gratitude to the jury members - Mme. Myriam Desainte-Catherine (LABRI, University of Bordeaux), Mme. Anne Warlaumont (University of California, Los Angeles), M. Richard Hahnloser (ETH, Zurich) and M. Jean-Luc Schwartz (CNRS, France) - for graciously agreeing to take part in the review and examination process of this thesis. Given your experience, I believe your feedback will be a valuable source of improvement and discussion for this work.

I would like to thank Inria Bordeaux Sud-Ouest and the Institute of Neurodegenerative Diseases (IMN) for welcoming me during the past three years. A special thanks goes to Chrystele, for guiding me through all the bureaucratic practices typical of new beginnings, and more. IMN has been a welcoming working environment, full of educational insights thanks to the multidisciplinarity that distinguishes it. I would like to thank Inria for the CORDI-S PhD fellowship grant and LabEx BRAIN for the PhD extension grant. Without their funding, this project would not have been possible.

I wish to thank all people whose presence was a milestone in the pursuing of this thesis.

The entire, current and former, Mnemosyne team. It's been a pleasure sharing the office with each of you, some for longer and some for less. I value not only the opportunities for learning and the scientific discussions, but also the coffee breaks and the convivial moments. Sharing the office with you has always been a pleasure, and never a duty.

A special thanks to Nathan, formerly an intern and now an engineer of the team, with whom I worked side by side in the last year of my PhD. Your dedication and perseverance made the completion of this project possible. Invaluable were the discussions to look for new ideas and the suggestions on how to represent something in a clear straight-forward way.

A heartfelt thanks to my best Mnemosyne-related people - Anthony, Bhargav, Cassio, Ikram, Remya, Snigdha and Thalita - for being a source of encouragement in this journey by giving me countless practical survival tips and many many many moments of lightheartedness. I am grateful to have found in you excellent colleagues, but, above all, precious friends.

Alongside Mnemosyne team, I would like to thank all the members of Arthur's team, for all the insights I was able to benefit from being able to interact with neuroscientists, delving into topics little known to me. Working side by side with those who do the experiments

has helped me to treasure their invaluable importance and the dedication they require.

I would like to thank the members of my PhD committee - Mme. Myriam Desainte-Catherine (LABRI, University of Bordeaux) and Pierre-Yves Oudeyer - for kindly accepting to follow me from a distance during these three years of PhD, for allowing meetings that have been a source of inspiration and new ideas.

Although the following people are not directly related to my research, they have played a major role in my well-being over the past few years both in Bordeaux and in Italy.

I wish to aknowledge the support of my family, my mother, Marta, and my father, Gianni. You have always believed in the value of education and opportunities. Thank you for being able to let me go as many times as I wanted to. I know that departures and not being able to be in range were not things that were easy for you to accept, but between a "Buongiorno" and a few calls to give IT support, you made it.

I would like to thank my sister, Anna, for being supportive from a distance during all these years, seeing a chance when I was lost in my worries not seeing a positive perspective.

I would like to thank everyone in my family, for the open-armed welcomes each time I returned, for visiting me in my new home, and most importantly, for always making me feel like part of the family, no matter where I am.

A loving thank to my boyfriend, Tellington, for believing in me every day, even when I don't believe enough myself. You never stopped believing that I could do it, not even when, caught up in my discouragement you would say, "Okay, then let it go." You knew I wouldn't let it go. That I would bang my head against it again and again. But, quietly and lovingly, you always knew how to respect my time.

A big thank to those who have made Bordeaux a place to call home.

"Un grazie disagiato" to Cristina and Giuliano, for being always present friends. Thanks to Cristina, my dearest coloc. It was a pleasure to share with you not only the PhD journey but also the space of daily life, Aristide33. You have always made me feel at home and have been family to me in Bordeaux. I treasure this experience, certain that I have found more than just any friendship, but rather a friendship with solid foundations and a lot more to come. Thanks to Giuliano, who always reminds me to think big and plan ahead, which I sometimes tend to forget. I value all the time spent together discussing about movies, books, politics, and more. It is, and it will always be, a pleasure to spend such a time with you.

Thanks to the Italian-related crew, for being a great family of friends in which to find support and a source of good cheer. Bordeaux would not have been the same without the delays at Nadia and Giovanni's, the wisdom of Gabri, the end-of-summer barbecues at Sally and Ale's, the sangria parties at Gloria and Ale's, the parties/evenings/dinners/aperitifs/whatever at the chateaux of Cecile, Sergio and, by adoption, Luigi. Thanks to those friends - Alessandro, Maya, Giuliana, Johannes, Laia, Federica, Costanza, Emanuela - who passed by, some for longer, some for less, leaving me a positive imprint and a life story to treasure.

May and Nuria, you've been among the first friends I had in Bordeaux and, step by step, we made this journey until the end. It has been a pleasure to share dinners, lunches, coffee breaks and travels (more to come) with you.

A special mention goes to my favorite soccer players. Girls, you have been the source of so many wonderful moments of venting and entertainment Playing soccer with you has been a pleasure experienced at every practice and game. Et pour le Spuc...!

Last but not least, a heartfelt thanks to "my people" back in Italy, for having followed my French chapter from a distance. I am grateful for all those relationships that, together, we were able to cultivate and nurture, even though we were far apart. Your friendship has been a heartening support.

To everyone who believed in me, especially my family.

Key Abbreviations

AEVB	Autoencoder Variational Bayes
aST	anterior Striatum
aT	anterior Thalamus
BMU	Best Matching Unit
CMA-ES	Covariance Matrix Adaptation - Evolution Strategy
CNN	Convolutional Neural Network
COSMO	Communicating Objects through Sensorimotor Operations
D	Discriminator Model
DIVA	Direction Into Velocities of Articulators
DLM	thalamic nucleus DorsoLateralis anterior par Medialis
ESN	Echo State Network
FF NN	Feed Forward Neural Network
F0	Fundamental frequency
FM	Forward Model
G	Generator Model
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GP	Gradient Penalty
HPF	High Pass Filter
IAC	Intelligent Adaptive Curiosity
IM	Inverse Model
IPL	IntraParietal Lobule
IS	Inception Score
LDA	Linear Discriminant Analysis
LFP	Local Field Potential
LMAN	Lateral Magnocellular nucleus of Anterior Nidopalllium
LMC	Laryngeal Motor Cortex
LPF	Low-Pass Filter
LTD	Long-Term Depression
LTP	Long-Term Potentiation
LP	Liepshitz Penalty
MFCC	Mel-Frequency Cepstral Coefficients
MN	Mirror Neuron
MNS	Mirror Neuron System

MSE	Mean Squared Error
NN	Neural Network
O	Optimization algorithm
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
qTA	quantitative Target Approximation
RA	Robust nucleus of Arcopallium
RBF	Radial Basis Function
RL	Reinforcement Learning
RNN	Recurrent Neural Network
S	Supervised Learning
SGVB	Stochastic Gradient Variational Bayes
SMA	Supplementary Motor Area
SMP	Song Motor Pathway
SOM	Self-Organized Map
SSE	Sum of Squared Error
STRF	Spatio-Temporal Receptive Field
STS	Superior Temporal Sulcus
SVM	Support Vector Machine
U	Unsupervised learning
UMAP	Uniform Manifold Approximation and Projection for Dimension Reduction
VAE	Variational Autoencoder
VLAM	Vocal Linear Articulatory Model
VTL	Vocal Tract Lab

Contents

General Introduction

Humans, like songbirds, learn by imitation from an early age (e.g. speech learning in children and song learning in songbirds): they are capable to reproduce an experienced sensory stimulus (e.g. a sound) by finding the appropriate motor command of reproducing it. The behavioral studies and the neuroanatomical structure of the vocal control circuit in humans and birds provide the basis for a bio-inspired model of vocal learning. In particular, birds have brain circuits exclusively dedicated to song learning, making them an ideal model for exploring the representation of vocal learning by imitation.

How to approach the definition of a sensorimotor learning model? What are the main components needed to build a minimal representation? This thesis aims to build a bio-inspired model to describe imitative vocal learning in birds using recent developments in artificial intelligence (e.g. Generative Adversarial Networks, the so-called GAN). An extensive review work has been conducted to understand deeply the fundamental concepts on which the thesis is based. **Chapter 1** contains a gentle introduction to these concepts: sensorimotor learning, vocal learning from both a biological and computational point of view and generative models. **Chapter 2** describes the representation of a minimal vocal learning model and how it has been implemented in literature. The comparison between several studies has been carried out carefully and the differences/similarities have been extensively discussed. Two tables contain the details of the comparison.

The first model proposed is a theoretical inverse model based on a simplified vocal learning model where there is no sound production. Such a simplification allows to investigate how to introduce biological assumptions into a vocal learning model and which pa-

rameters influence the computational power of the model the most. **Chapter 3** describes the architecture of a simple vocal learning model where a non-linear sensory response activates perceptual units and a normalized Hebbian learning rule drives learning. The influence of the sharpness of auditory selectivity and of the motor dimension is discussed.

In order to have a complete model (i.e. able to perceive and produce), a motor control function capable of reproducing sounds similar to real data (i.e. recordings of canaries and zebra finches, adults and juveniles) is needed. Latent models allow to obtain a low-dimensional representation of the sound, called latent space. **Chapter 4** explores the application of a GAN to a dataset of canary syllables. On the one hand, the study shows the capability of the generator of reproducing realistic syllables after training. On the other hand, it highlights the possibility of using a low-dimensional latent space.

The generator introduced above has been used as a motor function to build a sensori-motor model with an action-perception loop, where the connections between perceptual space and motor space are learned via an inverse model. **Chapter 5** describes the architecture of the vocal learning model and its components. Early phases of learning allow the investigation of whether or not the learning tends toward the target when using random motor exploration and a non-normalized Hebbian learning rule.

Chapter 1

Introduction

Contents

1.1 Introduction to sensorimotor learning

Humans show an innate ability to learn motor skills both in early and advanced stages of life. For instance, babies learn how to pronounce a word, or how to play with a toy. Similarly, adults can learn how to pronounce a word from a foreign language, or how to play an instrument. Sensorimotor learning allows humans to improve their ability to produce sensory-guided motor actions. Learning how to reproduce a sound or a movement requires processing sensory information, choosing the adequate learning strategy, selecting a target sound or movement (Wolpert et al., 2011).

Section 1.1.1 describes human behavior during sensorimotor learning, defining the concepts of imitation and trial-and-error strategy. Section 1.1.2 gives a general overview of brain areas involved in sensorimotor learning. Section 1.1.3 defines Mirror Neurons (MNs) and their implication in sensorimotor integration and learning. Section 1.1.4 describes the different theoretical frameworks developed to model sensorimotor learning.

1.1.1 Behaviour

During sensorimotor learning, a learner tries to execute actions and receives external feedback which gives information about whether the action is performed correctly or not. In the case of imitative learning, the learner is guided by a tutor. For instance, a parent teaching his child to pronounce a new word, or a coach showing his team how to kick the ball correctly. In the first case, the sensory stimulus is mainly an auditory stimulus (how a syllable sounds), even if a visual component could be present (e.g., the lips movement). In the second case, the stimulus is mainly visual (the actual demonstration of kicking the ball), sometimes enriched by a cognitive component (the vocal explanation that the coach might give). When learning a motor skill, humans focus on the relevant sensory

modality. When learning a new sound, a baby tries to reproduce the auditory stimulus. When trying to kick a ball, a player tries to place his foot in a similar position as his coach.

Imitation is a key mechanism for sensorimotor learning. It represents the ability to observe and replicate another's behavior to either learn new actions or to adapt to a changing environment. It is motivated by intention and purpose, such as curiosity (Zentall, 2001). Such motivation could come from external or internal factors. For instance, a child doing his homework might be motivated either by the fact that his parent won't bother him later (external motivation), or by the fact that he thinks that it will be good for his future (internal motivation) (Oudeyer and Kaplan, 2009). As shown in Figure 1.1, in both cases the action selection actuated by the learner (here called *autonomous agent*) depends both on the sensations derived from the external environment and on the rewards obtained. In the case of externally motivated behavior, the reward directly comes from the external environment. For instance, the parents' appreciation (or the absence of punishment) with respect to the child doing his homework. In the case of internally motivated behavior, the sensations coming from the external environment also provide input to the motivation system, which generates an internal reward and thus contributes to the action selection. For instance, the child's feeling of learning something useful to become an astronaut (here, the external motivation is given by the aim to become an astronaut). A third possibility is that the child is doing his homework because he is having fun: this is an intrinsically motivated behaviour (Oudeyer and Kaplan, 2009). Intrinsic motivation is internally driven by curiosity, novelty, knowledge and surprise.

Imitation relies on the ability to produce a motor command that replicates a previously experienced sensory stimulus. This ability is enabled by a causal relationship between the external stimulus and the produced action (in the case of a baby learning how to speak, the replication of a sound) (Kuhl, 2004). Once the observer has received the sensory stimulus, he tries to replicate the perceived behavior and to produce a motor command that enables the production of a response as similar as possible to the experienced sensory stimulus. Imitation behavior is not specific to humans: several examples of animals

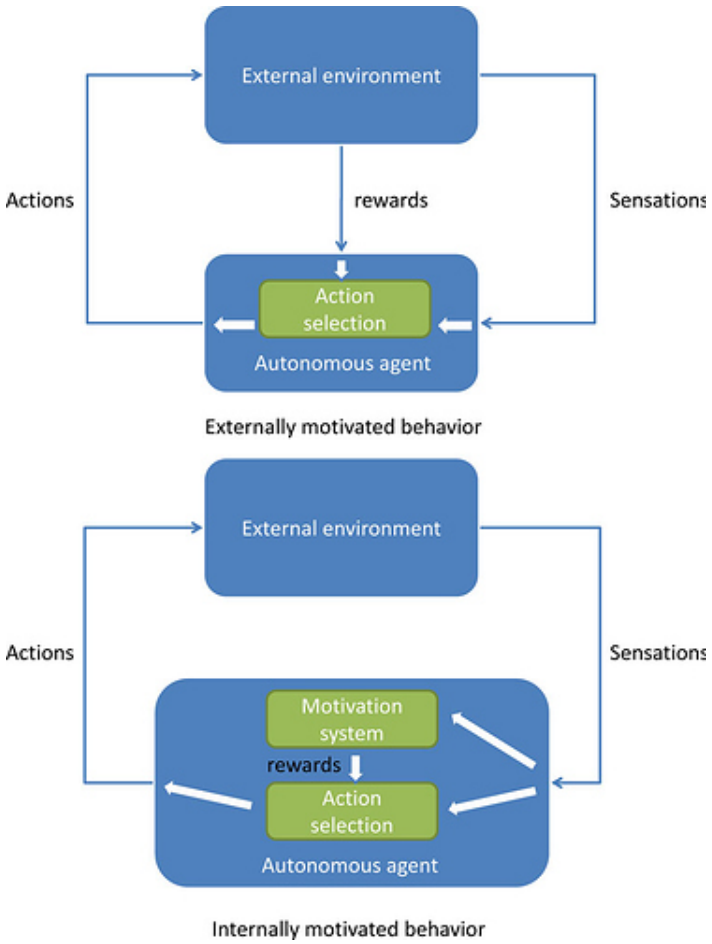


Figure 1.1: **External and internal motivated behavior.** Each learner can be driven by external (top panel) or internal motivation (bottom panel). The selection of the action is guided by the sensations coming from the external environment and a reward. The latter either comes directly from the external environment (in the case of external motivation) or is processed through a motivation system (in the case of internal motivation). Image from Oudeyer and Kaplan (2009).

showing imitative learning have been highlighted in the literature (Zentall, 2003). For instance, young canaries learn how to sing by copying the song of an adult tutor and trying to reproduce it until they have learned it (Brainard and Doupe, 2002). Or, home-raised chimpanzees learn how to wash the dishes and how to apply lipstick on their mouth, showing imitative learning similar to humans (Hayes and Hayes, 1952).

As an alternative to imitation, a trial-and-error strategy has first been described by (Thorndike, 1898) who observed that learning is promoted by positive results. Trial-

and-error is characterized by repeated attempts to reach a designated goal, until success. Each attempt is usually called *trial* and learning is characterized by a gradual improvement in performance (Krakauer and Mazzoni, 2011). The repetition of the successful action translates in an increased probability to perform this particular action (Verstynen and Sabes, 2011).

In no case is an animal activity to be interpreted in terms of higher psychological processes if it can be fairly interpreted in terms of processes which stand lower in the scale of psychological evolution and development (Morgan, 1903). Morgan's Canon claims that animal behavior should be explained in the simplest possible way. Behind the learning behavior, there are many psychological and neural processes involved in the sensorimotor system. The ability to estimate the external world and the state of our body allows the brain to process two types of information: a sensory prediction of the expected outcome, and a sensory feedback describing the actual outcome (Shadmehr et al., 2010). The sensory feedback provides pieces of information from sensory receptors through the afferent pathways. It can be provided by auditory or visual stimulation in response to a specific behavior (the perceived stimulus, i.e. the goal of the learning). Moreover, it contains information about the prediction error, which can support learning by evaluating the current output and driving the next trial (Mooney, 2009; Wolpert et al., 2011). The sensory prediction has the advantage of being available before the motor action takes place, and of overcoming the sensory feedback delay, and helps in measuring the environment (Shadmehr et al., 2010). Combining sensory prediction and sensory feedback, the perception would be more accurate. The prediction error on one trial is the error obtained by comparing the predicted outcome with the actual outcome and can influence the next trial (Thoroughman and Shadmehr, 2000; Tanaka et al., 2009). Computationally, the sensorimotor system estimates the motor command and computes an error signal (Wolpert et al., 2011) that guides the learner toward the correct motor command. This process is at the basis of Reinforcement Learning (RL), as described in Section 1.1.4.

1.1.2 Brain circuits

Sensorimotor learning involves several areas of the brain that are related to the perception of a stimulus and to motor production. Imitation relies on sensory areas to process the sensory information (i.e., the information coming from the sensory stimulus while perceiving it) before performing the movement. Sensory information is first processed through sensory ascending pathways (from the receptor organ up to cortical sensory areas). Such pathways are classified according to their functional components and their anatomical location, and convey the sensory information to the thalamus, the cerebellum and the prefrontal cortex. Then, motor descending pathways convey the information from the cortical sensory areas to the muscles: such pathways are involved in motor control. Along with motor cortical areas, the prefrontal cortex, the basal ganglia and the cerebellum play a role in skill learning (Krakauer and Mazzoni, 2011). The prefrontal cortex plays a role in task strategy, while the basal ganglia contribute to optimizing the motor output. The basal ganglia circuits indeed promote sequence learning through trial-and-error learning and help the neural system promoting optimal motor and cognitive control (Graybiel, 2005). Moreover, basal ganglia contribute to the generation of a long-term representation of learned actions, supporting the notion that it participates in learning and adaptation (Bédard and Sanes, 2011). Basal ganglia are linked to reinforcement learning (Doya, 2000), whereas the cerebellum to linked with supervised learning (and, in particular, internal models). Functionally, cerebellum and basal ganglia are slightly different: they participate in motor control with specific functions (Graybiel, 2005). Basal ganglia are involved in the specification of the movement before execution (selection, preparation, retention), whereas the cerebellum is more involved in setting the movement parameters during execution (Jueptner et al., 1997). Furthermore, the cerebellum has a crucial role in adaptation (Mazzoni and Krakauer, 2006; Galea et al., 2011; Wolpert et al., 2011), i.e. its contribution in adjusting the learning model (using the prediction error) can result in faster adaptation (Krakauer and Mazzoni, 2011). Once learning is completed, the motor skills representation is encoded by the motor cortex (Gentner and Classen, 2006; Reis

et al., 2009).

The encoding of a sensory stimulus reveals premotor circuits helping in processing the perceived stimulus (Rizzolatti et al., 1996; Roberts et al., 2012). During imitation, regions in the inferior frontal cortex and inferior parietal cortex are activated (Iacoboni et al., 1999). When an action is retrieved, even if not executed, it may play a role in understanding the motor events. That is, there exists an observation/execution system, which is coherent with the findings of mirror neurons in macaque monkey prefrontal cortex (Rizzolatti et al., 1996; Rizzolatti and Craighero, 2004).

1.1.3 Mirror neurons

Mirror neurons (MNs) are neurons that fire both during the observation and the production of a gesture (Gallese et al., 1996; Oztop et al., 2013). Due to this interesting property, mirror neurons have been largely investigated for the importance they could have in social neuroscience (Ferrari and Rizzolatti, 2015), imitative learning(Cross and Iacoboni, 2014) and culture evolution (Tramacere and Moore, 2018).

Mirror neurons (MNs) have been discovered in the premotor cortex and Intraparietal Lobule (IPL) of macaque monkeys (Di Pellegrino et al., 1992; Rizzolatti et al., 1996). IPL plays a role in action organization and action understanding: it's part of the motor system. IPL and the Superior Temporal Sulcus (STS), an area that codes behaviors such as walking and hand movement, form the Mirror Neuron System (MNS). STS does not contain mirror neurons but is involved in the process of information about action observation and execution. Lately, mirror neurons presence has also been highlighted in the medial frontal cortex known as supplementary motor area (SMA) (Yoshida et al., 2011).

Moreover, mirror neurons have been found in HVC (used as a proper name), a brain area responsible for song perception, learning and production (Prather et al., 2008). Neurons from HVC projecting to other brain areas of the bird exhibit patterns of sing-related and auditory activity. The former could provide a motor estimation of auditory feedback,

the latter could facilitate communication (Prather et al., 2008).

Although the investigation of mirror neurons activity primarily relies on electrophysiological recordings in monkeys and birds, techniques such as neuroimaging and magnetic resonance can find mirror-like activities in humans. A mirror system for speech perception and production has been identified (Pulvermüller and Fadiga, 2010), and audio-vocal mirror activity has been observed in human Broca's area (Wilson et al., 2004), the homolog region of monkey ventral premotor cortex (Gallese et al., 1996). Also, these techniques have been used to study mirror responses associated with the perception and the execution of facial or hand gestures in humans (Ferrari et al., 2003; Rizzolatti and Craighero, 2004; Heyes, 2010). Here, the coupling between observation and execution of the movement in a social context has been observed. That is, MNs may have a role in social learning (Tramacere and Moore, 2018; Arbib, 2005).

1.1.4 Theoretical frameworks for sensorimotor learning

Depending on the type of information used during learning (e.g., prediction of the outcome gesture), different learning processes can be distinguished (Wolpert et al., 2011). Internal models (IM) are predictor models that can be used to understand sensorimotor learning and motor control behaviour (Wolpert et al., 1995). In particular, they propose a mechanism to improve motor control, defining the relationship between an experienced sensory stimulus and the motor configuration needed to reproduce that exact stimulus. Moreover, imitative learning, where a learner tries to replicate (by finding the correct motor configuration) the tutor's gesture, can rely on internal models. There are two types of internal models: inverse models and forward models[1], that can be used one at a time, or coupled. Inverse models estimate the motor command that corresponds to the desired state. Forward models predict the expected sensory feedback of a motor command. Moreover, a forward model can be used to integrate a prediction error able to predict the sensory consequences of a motor action (Wolpert et al., 2011). If both are present,

[1]In contrast with internal forward models, goal babbling literature call forwards models as the real forward model that is given by the world (Rolf et al., 2010; Rolf, 2013; Reinhart, 2017; Philippsen, 2021).

the model describes the learning of a bi-directional mapping between motor and sensory variables.

Neurophysiology of MNs is compatible both with an inverse model and a forward model (Tramacere et al., 2019). MNs could be part of an inverse model between the sensory representation of intended actions and the motor commands to reach them (Hanuschkin et al., 2013). Alternatively, MNs are also compatible with a forward model: the internal prediction generated by auditory feedback and tutor perception may be learned by associating the auditory feedback registered in HVC and the premotor activity in birds (Tramacere et al., 2019).

Reinforcement learning (RL) is a basic machine learning paradigm (along with supervised learning and unsupervised learning), that formulates how learning can occur by taking actions following a certain criterium (i.e., a policy) and receiving a reward consequently. The amount of reward received for a given action is given by the value function. RL can be divided into two categories, depending on how the policy and the value function are estimated: model-free RL and model-based RL. Model-free algorithms can be thought of as a direct implementation of trial-and-error method (Sutton and Barto, 2018): the algorithm does not use the prediction of the environment and the reward function. Alternatively, a model of the environment and the reward function are used to estimate the optimal policy in model-based algorithms. Recently, other classes of RL algorithms have been proposed. Inverse RL substitutes the reward with information coming from an external observer (Ng et al., 2000). Deep RL extends RL using deep neural networks (François-Lavet et al., 2018): this approach allows to solve complex tasks requiring high-dimensional inputs. In RL the agent must explore different possibilities to be successful and to improve the motor output (by improving the motor commands): this comes from the fact that the received information about how good is the motor output, and not directly information about which is the correct direction it should take (which would be the case in supervised learning) (Wolpert et al., 2011). This progressive improvement of the produced gestures provides a theoretical framework to explain the mechanisms under trial-and-error learning (introduced in Section 1.1.1).

Sensorimotor learning requires exploration mechanisms, which can be used to drive learning. Random motor exploration is the simpler exploration strategy (so-called *motor babbling*) (Rohde et al., 2019). More complicated exploration strategies are goal-directed and could be driven by intrinsic or extrinsic motivation. A goal-directed strategy takes into account the memory of the current and preceding perceived outcome (Rolf et al., 2010). It could be based on the novelty level of the goal or on the competence of the agent. For instance, a *goal babbling* approach defines an exploration strategy constantly motivated by goals and not simply random.

1.2 Vocal learning

Vocal learning represents the ability to acquire new sounds via imitation, and should not be confused with *usage vocal learning* and *auditory learning* (Petkov and Jarvis, 2012). The former involves the learning of the context in which to vocalize, and not the production of new vocalizations itself. For instance, a predator alarm calls, or another appropriate behavioral response upon hearing the call from a conspecific (Seyfarth et al., 1980). The latter represents the ability to form memories of experienced sounds (even if these sounds are not included in the learner's innate repertoire) and to react differently to different sounds. For instance, a dog that performs the act of sitting after the human says *sit*.

Several animals, like parrots, songbirds, bats, elephants, dolphins, and humans can produce vocalizations (sounds generated by the vocal organ) and learn by imitation from a tutor (Petkov and Jarvis, 2012). That is, they show vocal learning in their behavior. For this reason, they are identified as *vocal learner*, in opposite with *vocal non-learner*. The distinction between vocal learners and vocal non-learners evidences different levels of vocal learning ability (Petkov and Jarvis, 2012): vocal non-learners have a lower level of laryngeal (mammalian) or syringeal (avian) control as seen in complex vocal learners. For Janik and Slater (1997) complex vocal learning is determined by whether or not an animal can learn to copy a sound from another species. When animals copy complex

vocalizations not present in the species-specific repertoire they are forcibly learning a new acoustic template, and then learning how to develop a matching vocalization. These hypotheses suggest the presence of different levels (non-learners, limited vocal learners, moderate vocal learners, complex vocal learners and high vocal learners) of vocal learners and the hypothesis that vocal learning is a continuum rather than a binary classification (Arriaga et al., 2012). For instance, humans are considered high vocal learners, and songbirds are classified as complex learners. Non-human primates, mice, and goats are considered limited vocal learners under this system (Petkov and Jarvis, 2012; Tyack, 2020).

From a neuroanatomical point of view, brain control circuits in animals show differences depending on their behavior (vocal non-learners and learners), or order (birds, mammals, etc.). The auditory-motor pathway connectivity of vocal non-learners and learners shows differences: for instance, the auditory pathway of a vocal non-learner does not provide input to the vocal learning system (Petkov and Jarvis, 2012). This comes from the fact that vocal non-learners rely on an innate vocal-production system where auditory input is not required to regulate vocal production. On the contrary, vocal learners provide an auditory input that enables the learning of complex motor control.

In birds and humans, brain circuits involved in vocal learning contain two main pathways: the vocal learning pathway and the vocal production pathway Chakraborty and Jarvis (2015). The vocal learning pathway is responsible for vocal imitation and plasticity. The vocal production pathway is responsible for vocal production. Section 1.2.1 and Section 1.2.2 respectively introduce vocal learning in humans and in birds, highlighting the behavior paradigm, and the main brain circuits that underline the learning process. In particular, Section 1.2.2 clarify why birdsongs are a powerful model to study vocal learning. Section 1.2.3 defines the acoustic of birdsong, focusing on the important features and the available tools to deal with segmentation and features extraction. Section 1.2.5 briefly introduces the structure of a minimal computational model for vocal learning. An extensive and detailed review of vocal learning can be found in Chapter 2.

1.2.1 Vocal learning in humans

Behavior

During their first year of life, infant speech-perception and speech-production develop in parallel. Speech-perception is characterized by a sensory learning phase, which begins immediately after birth when infants experience their first exposure to vocalizations. Starting from their sixth month of life, infants begin to develop language-specific vowel perception, and it's only around the first year of their life that they can perceive native language consonants (Kuhl, 2004). During all their learning period, infants are exposed to vocalizations and guided by auditory feedback that guides the learning. At birth, and during the first three months of their life, infants produce non-speech sounds called *protophone* (e.g., quasivowels and primitive squeals), which are considered to be the earliest precursors to speech (Oller, 2000; Buder et al., 2013; Oller et al., 2019; Warlaumont, 2020). During this period infants learn how to control infrastructural speech properties (Oller, 2000). Within the third and the seventh month of their life, infants learn how to control their vocal tract and begin to produce vowel-like sounds with a significant variation in duration, amplitude, and quality (Kuhl, 2004; Oller et al., 2013; Warlaumont, 2020). Moreover, they become able to produce consonant-like elements and to add pauses in phonations (Oller and Eilers, 1988; Oller, 2000). At approximately seven months of age, infants begin to produce canonical babbling (Kuhl, 2004; Buder et al., 2013; Warlaumont, 2020) i.e. a canonical syllable without a clear meaning. A canonical syllable is a vocalization containing at least one full vowel coupled with a following or preceding consonant (Warlaumont, 2020). Interestingly, even when infants reach the ability to produce meaningful speech, canonical babbling continues to be present in their production, and gradually decreases (Robb et al., 1994). Around the first year of their life, infants begin producing words (Kuhl, 2004; Warlaumont, 2020), which at the beginning could be better understood by their caregivers (Baudonck et al., 2009). Then, infants develop their vocabulary, adding a richer and more flexible lexicon (Warlaumont, 2020).

Brain circuits

Diving in the neuroanatomical framework, a cortico-basal ganglia-thalamocortical loop involves connections between the cortex, the basal ganglia, the thalamus, and back to the cortex. Thus, the vocal learning pathway in humans includes the Broca's area, the Wernicke's area and superior temporal gyrus (Friederici, 2011; Jarvis, 2019). The vocal production pathway primarily involves the laryngeal motor cortex (LMC), from where the connections to vocal motor neurons start (Chakraborty and Jarvis, 2015; Jarvis, 2019). That is, the motor cortex is the core area responsible for controlling movements in the human brain. A more precise description of the neuroanatomy of the human brain can be found in Section 2.2.2 of Chapter 2.

1.2.2 Vocal learning in birds

Songbirds represent the most studied model organisms of vocal learning for many reasons. First, birds share with humans not only a similar behavior paradigm during vocal learning (Kuhl, 2004), but they have also a comparable neuroanatomical structure (Chakraborty and Jarvis, 2015). Besides, in songbirds brain structure there is a circuit dedicated exclusively to vocal learning, which gives inputs to study and to understand human brain structures. Last but not least, songbirds are easier than bats or cetaceans to deal with from an experimental point of view.

Behavior

Vocal learning in birdsongs is characterized by two distinct phases, **sensory learning** (related to sound perception) and **sensorimotor learning** (related to sound production) (Kuhl, 2004). At the beginning of their life, juveniles birdsongs listen to a tutor and they do not produce any vocalization. Sensory learning begins immediately after hatch and lasts until the juvenile birds start to produce earlier songs. During this period, the pupils memorize their tutor songs (Mooney, 2009). It's only later that the birds start to produce their song. Three phases characterize learning production in birds: subsong,

plastic song, crystallization (Brainard and Doupe, 2002). During sensorimotor learning juvenile birds relies on auditory feedback to match their own song to the memorized model (Mooney, 2009). A variable babbling behavior, called **subsong**, characterizes this phase of learning. Subsong vocalizations are driven by an immature motor pathway and become more and more plastic and identifiable with time. To do so juvenile birds adapt their vocalizations to incorporate some elements of the tutor song. Finally, the plastic song is gradually transformed into highly complex, stereotyped motifs (Aronov et al., 2008), becoming less and less dependent on the auditory feedback (Mooney, 2009). This phase is called **crystallization**. Birdsongs are now able to produce stable adult vocalizations, highly similar to the tutor song.

The degree of vocal learning vary within species: songbirds like hummingbirds learn only during a single sensorimotor phase of learning, other songbirds like canaries show several sensorimotor phases of learning during which they can learn new sounds (Petkov and Jarvis, 2012). This difference separate birdsongs into two categories: closed-ended learners, and open-ended learners (Brenowitz and Beecher, 2005). Closed-ended learners such as the zebra finches and hummingbirds can only learn during a limited period and subsequently produce highly stereotyped or non-variable vocalizations consisting of a single, fixed song which they repeat their entire lives. In contrast, open-ended learners, including canaries and various parrot species, display significant life-long plasticity and continue to learn new songs throughout their lives.

Brain circuits

Neuroanatomy of birdsong involves auditory and motor areas Chakraborty and Jarvis (2015); Jarvis (2019). The vocal production pathway projects from HVC, to the robust nucleus of arcopallium (RA), which is analogous to the laryngeal motor cortex (LMC) in humans. RA controls the syrinx in birds, similarly as LMC controls the larynx in humans. The vocal learning pathway involves indirect projections from the song-related basal ganglia nucleus Area X to the robust nucleus of arcopallium (RA), the lateral magnocellular

nucleus of the anterior nidopallium (LMAN), and a thalamic nucleus. This pathway is responsible for imitation and plasticity, forming a basal-ganglia-thalamocortical loop. A more precise description of these pathways and a figure helping the comparison between the neuroanatomy of the brain in humans and birds can be found in Section 2.2.2 of Chapter 2.

The basal-ganglia-thalamocortical loop is highly selective for the bird's own song (Solis and Doupe, 1997). On the one hand, auditory neurons in adult zebra finches respond more to the bird's own song than to songs from other conspecifics. On the other hand, juveniles' neurons respond well to both their own song and the tutor song (from conspecific adults). The sensory properties of Area X and the Lateral Magnocellular nucleus of Anterior Nidopallium (LMAN), two of the basal-ganglia-thalamocortical loop nuclei, in both young (the presence of auditory neurons in early stages of learning) and adult birds (selectivity to bird's own song) are consistent with an auditory role of the loop in the song-learning process (Doupe, 1997).

1.2.3 Structure of speech and birdsong

Human speech and birdsongs consist of strings of sounds separated by silent intervals (Doupe and Kuhl, 1999). Analogies can be found in speech and birdsong. Infants, as speech development advances, start to produce vowel sounds, with or without surrounding consonants, i.e., *syllables*. Infants generally produce sounds having about the same duration as syllables in adult speech (Kent and Murray, 1982). That is sounds of length $400ms$ or less, with some exceptions that reach a duration of $2s$ or more. Differences in vocal tract features result in different acoustic feature distributions: for instance, because of the anatomical properties of their vocal tract, infants produce a more nasal sound with respect to adults (Kent and Murray, 1982). Only about their third month of life, infants begin to produce non-nasal vowel sounds (Oller, 1978), as a consequence of the vocal tract development.

In songbirds, the smallest unit of a song is the *note*. A note is a sub-syllabic element

and the combination of multiple notes originates a *syllable*. Syllables are stereotyped sounds which duration ranges from 20 to 200ms (Markowitz et al., 2013), and the repertoire is usually bird-specific (Lehongre et al., 2008). However, some syllables can be produced across families of the same species, such as canaries (Güttinger et al., 1978; Güttinger, 1985). Concerning canaries, when syllables are repeated multiple times, they form a *phrase* which can range from 500 ms to 3s of duration. There is no general correlation between the length of a phrase and the length of the syllables, but rather there is a correlation between the length of a phrase and the length of phrase that comes before or after (Markowitz et al., 2013). Finally, phrases are chained together to form songs of about $5 - 15s$ duration (Belzner et al., 2009). In zebra finches, a song is composed by a repeated sound pattern, called *motif* (Leonardo and Fee, 2005). The motif is composed of a sequence of syllables.

Although there is no evidence that the complexity of human language is observable in nonhuman animals (Beckers et al., 2012), there are similarities and parallels between birdsong and speech. The main difference between birdsong and speech is that in the former a simpler grammar is involved (Doupe and Kuhl, 1999). Another difference is that speech can convey complex meaning, which is not the case for birdsong. Nevertheless, birdsong can be compared with spoken speech (Doupe and Kuhl, 1999).

1.2.4 Sensorimotor integration in vocal learning

In the context of vocal learning, sensorimotor integration can be defined as the set of mechanisms connecting vocal production and vocal perception. For instance, the set of mechanisms connecting speech production and speech perception in humans, or song production and song perception in birds. The motor theory of speech perception states the role of the speech motor system not only to produce speech but also to detect it. The main hypothesis is that humans perceive speech by identifying the vocal tract gestures that produced it rather than by identifying the sound patterns generated by that speech (Liberman et al., 1967; Liberman and Mattingly, 1985). The motor speech the-

ory contains two main claims (Liberman and Mattingly, 1985). The objects of speech perception are the phonetic gestures of the speaker, represented in terms of the invariant motor commands needed to produce certain linguistic configurations. The motor theory by Liberman and Mattingly (1985) confirms that speech production and speech perception are both motor, and are regulated by the same structural constraints.

Such a link is supported by the existence of mirror neurons and of a common coding between the representations used for perception and action. Schwartz et al. (2012) proposed a perceptual-motor theory of speech perception, called Perception for Action Control Theory (PACT) of speech perception. Speech percepts seem to be related both to sound and to gestures. During speech production, auditory commands are non-linearly transformed into complex acoustic features. During perception, the motor system can be accessed and complex connections between perception and gestures arise. The role of perceptual-motor interactions is to predict missing information and to relate perceptual categories to their motor content.

There exist evidences of the influence of sensory perception on speech production and the influence of motor speech system on speech perception (Hickok and Poeppel, 2007). The perceptual system self-monitors the speech output via feedback signals and the auditory system plays an important role in speech production. For instance, speech production can be negatively influenced by the delay in the auditory feedback (Stuart et al., 2002), or by deafness (Waldstein, 1990), or by a shift in the voice pitch (Burnett et al., 1998). The fact that the motor speech system influences speech perception could explain the fact that there is not a one-to-one relation between acoustic patterns and perceived speech sounds (Liberman et al., 1967). Behind this idea, there is the fact that auditory signal can be variable across different sounds, whereas the motor configurations that produce them are often fixed. Moreover, it has been shown that the motor speech system is not necessary to solve the problem of context-dependency (Lotto et al., 2009). Rather, the auditory system maintains an estimate of the acoustic context and uses this information while encoding sounds. Recently, the modulation of perceptual response via motor-speech stimulation has been shown (Hickok et al., 2011). In the end, birdsongs and

humans behave optimally depending on the information they can gather from the motor plan and the auditory feedback (Hahnloser and Narula, 2017).

1.2.5 Modeling vocal learning

Vocal learning supposes the existence of a set of auditory targets that the learner tries to reproduce. Each target is an external sound that the learner processes and uses to guide his own sound generations. Thus, a required feature for a vocal learning model is the ability to produce and process a sound.

The representation of a minimal vocal learning model includes three feature spaces and the functional connections between them (Oudeyer, 2005). The motor space contains the motor parameters (related to the anatomical structure of the vocal organ). The sensory space contains the auditory stimulus (the real sound), which is generated by the motor control function. The goal space is the space of the perceptual representation or motor command corresponding to the sensory output that the bird wants to reproduce. Two types of models can be defined, depending on how the goal space is defined. In the sensorimotor model with an *action-perception loop*, the sensory stimuli are encoded via the sensory response function in the perceptual space (a low dimensional representation of the sound). This model potentially includes an inverse and a forward model between the motor space and the perceptual space. Alternatively, in the non-perceptual sensorimotor model, there is a non-perceptual representation of the goals (called internal representation). A goal-to-motor model learns the connection from the internal representation to the motor space. In this scenario, and depending on the learning framework, the sensory response function could provide a reward or an evaluation of the learning (Pagliarini et al., 2020). For a better understanding, refer to Figures 2.1 and 2.2 of Chapter 2: a schematic figure of each model highlights the architecture.

Motor control

To have a complete model (i.e., able to produce sounds), the aim is to have a motor control function capable of reproducing sounds similar to real data (e.g. recordings of canaries, vowels, sentences). The input set of the motor control function, called motor space, is the set of the motor parameters used to produce the sound (see Chapter 2). For instance, a set of parameters describing the tongue position in humans (Howard and Messum, 2007; Howard and Huckvale, 2005). The motor control function is a computational model of the vocal apparatus. For instance, in humans, it describes the joint activity of the respiratory system, vocal organs and vocal tract. Section 2.4 in Chapter 2 extensively detail how the motor space and the motor control function has been defined in the vocal learning literature.

Sensory function

As mentioned in Section 1.2.5, the sensory function results in a perceptual representation of the sound and, eventually, helps the evaluation of the produced sound. In the first scenario, the sensory function allows to encode the sound in a low-dimensional space and provides feedback of the motor command. In the second scenario, the sensory function contributes to the reward signal and can be used to drive learning towards a specific goal. Section 2.5 in Chapter 2 motivates the choice of the sensory response function, the sensory space and the perceptual space.

Learning architecture

One way to model the connection between the perceptual space (or the internal representation) and the motor space is using plastic synapses. The simplest artificial Neural Network (NN) is the feedforward neural network, where the connections between the nodes do not form a cycle. The information moves from the input nodes to the output nodes, passing eventually through hidden nodes, following one direction. The simplest example of a feedforward neural network is the perceptron: here, the inputs are directly

fed to the single layer of output nodes. Then, multi-layer perceptrons contain several layers, where each neuron in one layer is connected to all neurons in the next layer. Alternatively, Recurrent Neural Networks (RNNs) contain loop connections between the nodes and can use their internal state (called memory) to process sequences. This property makes them interesting to solve problems such as speech recognition (Li and Wu, 2015) or vocal learning (Hinaut and Dominey, 2013; Philippsen et al., 2014; Warlaumont and Finnegan, 2016). Although usually the learning architecture defined in a vocal learning model is based on a feed-forward neural network or a RNN, recent deep learning studies have highlighted several powerful tools that could be used to model vocal learning. Convolutional Neural Networks (CNNs) represent a famous extension of multi-layer perceptrons: the architecture includes an input and an output layer, as well as multiple hidden layers (LeCun et al., 1989). CNNs find origins in models of visual processing: the connectivity pattern between neurons is inspired by the organization of the animal visual cortex (François-Lavet et al., 2018).

Internal models and reinforcement learning can be used to learn connections between the perceptual and the motor layer, or to define a trial-and-error algorithm aiming to succeed in a pre-defined task. Section 2.6 in Chapter 2 highlights the different learning architectures that have been implemented to model vocal learning in literature.

1.3 Generative models

Generative models can generate new data samples, after having been trained on a dataset. For instance, a generative model could generate new samples of images containing flowers that look like real flowers. To be able to generate a new image of a rose that looks like a real rose, a generative model needs to learn the distribution of the real data. Formally, given a dataset X and a set of labels Y, a generative model estimates the joint probability $P(X, Y)$ (Ng and Jordan, 2002).

At first, generative models can be classified depending on the density function they define: it can be explicit or implicit (Goodfellow, 2016). Figure 1.2 shows a schematical

summary of generative models. For simplicity, the models have been thought as if they were implemented in their maximum likelihood formulation (that is, they aim to maximize the likelihood). Models that construct an explicit density could deal with a tractable function or with an intractable function. In the case of an intractable density, the use of approximations is required to maximize the likelihood. For instance, an approximative density can be obtained using a variational approximation, that is, using Variational Auto-Encoders (VAEs). Alternatively, there are models that do not represent explicitly a probability distribution, but rather represent a way to interact with it. For instance, the act of drawing samples from a distribution is an interaction between the generative model and the probability distribution of the input. This is the principle of Generative Adversarial Networks (GANs).

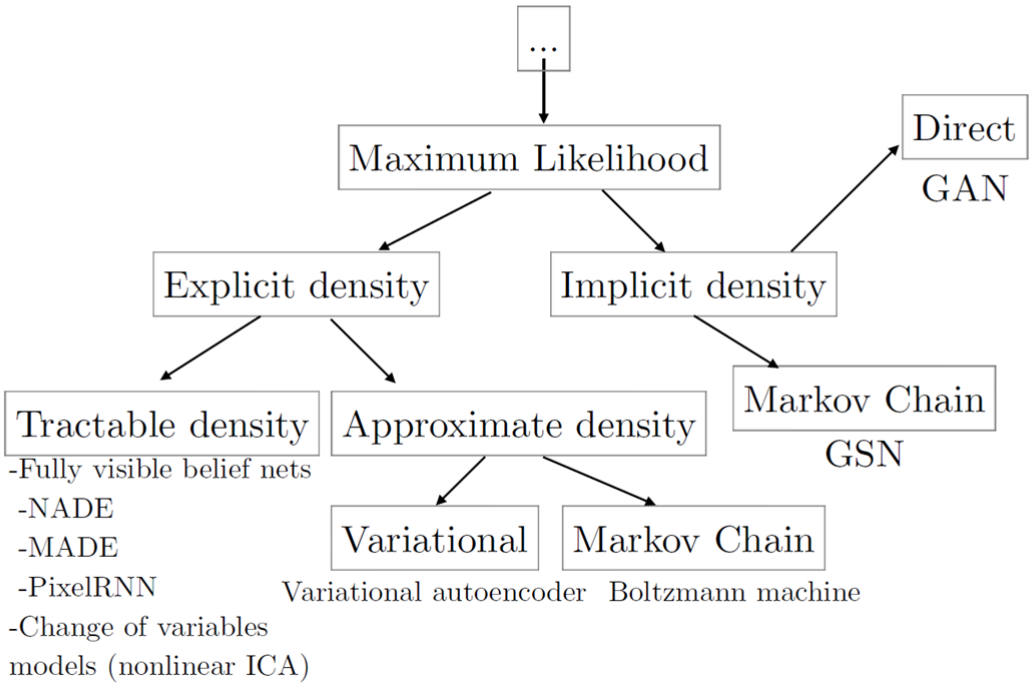


Figure 1.2: **Generative models taxonomy.** Generative models can define either an explicit or an implicit density. On the left branch, an explicit density can bifurcate into a tractable or an intractable density. The latter makes necessary the definition of an approximative density, that might be given by a variational approximation in the case of VAEs. On the right branch, an implicit density can be solved using a Markov chain or, in a direct way, by a GAN. Image from Goodfellow (2016).

Generative models are a great tool to deal with complex data and high-dimensional probability distributions in a wide variety of engineering domains (Goodfellow, 2016). For instance, generative models such as GANs and VAEs enable to represent high-dimensional distributions in a meaningful lower-dimensional representation called *latent space*.

Subsection 1.3.1 defines what are latent variables and latent models. Subsections 1.3.2 and 1.3.3 introduce, respectively, the basics of VAEs and GANs, highlighting the advantages and drawbacks. Subsection 1.3.4 describes the available models to generate sound, with examples both from the music domain and from the speech domain.

1.3.1 Latent models

Latent variables, or hidden variables, are variables that cannot be directly observed. Instead, they are inferred through a mathematical model from the observable variables. Indeed, latent variables are meaningful, but not directly measurable: they usually encode abstract concepts, such as categories or hidden data structures. Latent variables allow to reduce the dimensionality of the data, to compress the relevant information about the input data into a lower-dimensional space, and to understand better the manifold structure of the data.

The relationship between a set of observable variables (the dataset containing the real images of flowers) and a set of latent variables (a symbolic representation of the real data) can be described by a so-called latent variable model. Latent space models are capable of learning the fundamental characteristics of a training dataset and able to represent the variation of real data in a lower-dimensional space. When compressing the manifold of the dataset, latent space models tend to organize it based on fundamental qualities, which clusters similar examples close together (Roberts et al., 2018). Moreover, they can both reconstruct real examples with high accuracy and generate samples that are not in the training dataset. Indeed, it is possible to observe the latent space structure by using simple arithmetic and observing the outcoming generations. Figure 1.3 shows the analysis of the latent space structure made by Radford et al. (2015): it is possible to obtain

generations that are similar to the training dataset but do not belong to it. Moreover, with this analysis, it is possible to see an example of a conditional generative model, where generations can be manually tuned to show a particular feature (for instance, the presence or not of glasses).

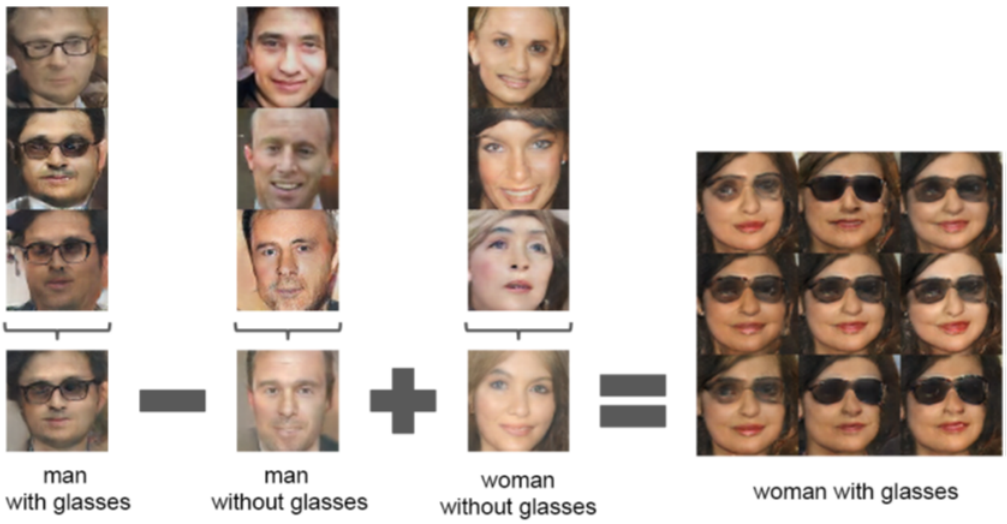


Figure 1.3: **Latent space structure.** The z samples corresponding to each image are averaged over each column in order to obtain an average latent vector representing a certain set of features. Then simple arithmetic produces the new latent vector, which has been used to generate the first sample on the right (top-left corner). Finally, noise has been applied to this latent vector to modify it and use it to generate 8 more images. Image from Radford et al. (2015).

1.3.2 Variational Auto-Encoders (VAEs)

VAEs are a particular instance of a more general algorithm, called Autoencoder Variational Bayes (AEVB) (Kingma and Welling, 2013). This algorithm is based on the

Stochastic Gradient Variational Bayes (SGVB) estimator that allows to efficiently approximate posterior inference.

The standard schema of a VAE as a graphical model is shown in Figure 1.4. A dataset of N i.i.d samples of continuous or discrete variables x is randomly generated from a set of variables z. VAEs assume that is not possible to give a simple interpretation of the dimensions of z: instead, z can be taken as a normal distribution with mean 0 and variance 1 (i.e., $z \sim N(0, 1)$) and can be interpreted as a latent code (i.e., a latent space) (Kingma and Welling, 2013; Doersch, 2016). The *encoder* and the *decoder* are probabilistic models. The former models the distribution over the possible values z from which samples x have been generated. The latter models the distribution over the possible values x, given the latent code z. These two probabilities rely on two parameters that are jointly learned with the Autoencoder Variational Bayes (AEVB) algorithm: the variational parameter φ and the generative model parameter θ (Kingma and Welling, 2013). A schema of a standard VAE represented as a graphical model is shown in Figure 1.4.

The learned model can be used for many tasks such as visualization, denoising or recognition. VAEs are composed of an encoder (recognition model) and a decoder (generative model). The encoder takes a real sample as input and builds a lower-dimensional representation of it (i.e., a latent space). The decoder takes a latent vector (i.e., an element of the latent space) as input and generates back the original sample. In this framework, the decoder represents the generative model (Doersch, 2016). Some authors (Goodfellow, 2016; Doersch, 2016) state that VAEs often obtain very good likelihood but produces low-quality samples (lower than GANs, for example).

1.3.3 Generative Adversarial Networks (GANs)

In the original formulation of GANs (Goodfellow et al., 2014), given a training dataset, two players (a generator model G and a discriminator model D) are involved in a back-and-forth competition. The generator model takes noise as input and aims to learn how to produce samples as much similar to the real samples as possible. The discriminator takes

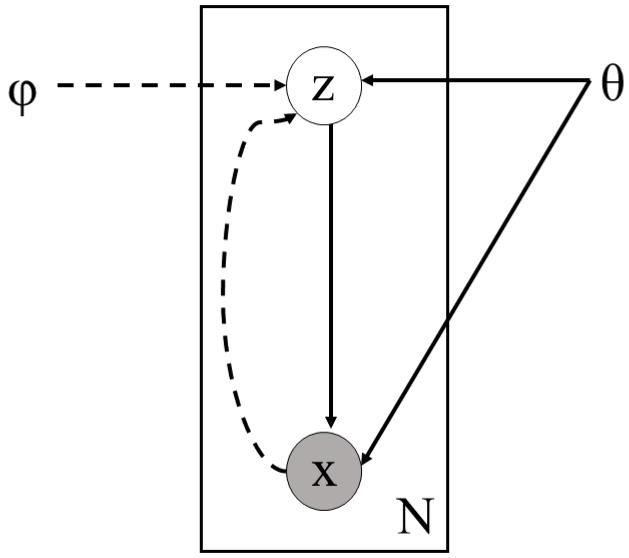


Figure 1.4: **Varietional Autoencoder general schema.** A standard VAE allows to sample from any input $z \sim \mathcal{N}(0, 1)$ keeping fix the generative model parameter θ. φ is the variational parameter. Image reproduce from Kingma and Welling (2013); Doersch (2016).

real and generated samples as input and aims to maximize its capacity of differentiating between them. That is, G and D are involved in a min-max game where G aims to minimize the loss function (defined using the Jensen-Shannon divergence (Goodfellow, 2016)), and D aims to maximize it. As summarized in the right part of Figure 1.5, G is trained several times, then D is used to obtain a validation of the training: it can output a value between 0 and 1, which represents the probability of whether a sample comes from the distribution of the data (a value near 1) or not (a value near 0) (Goodfellow, 2016). Of course, as shown in the left part of Figure 1.5, D applied to the real data should result in a value near 1.

In further developments of GANs, several authors proposed variations in the model architecture or in the loss function. More stable training has been obtained by using deep convolutional neural networks (Radford et al., 2015) or modifying the loss function definition. In particular, the loss function has been enriched by an additional term describing

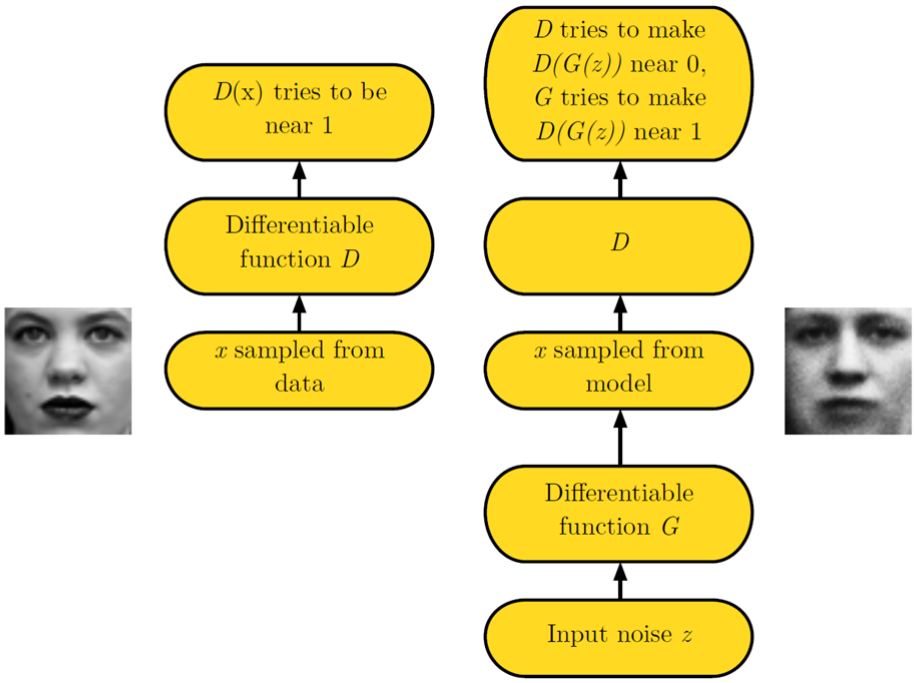


Figure 1.5: **GAN framework.** The first formulation of GANs included two models: a generator model G and a discriminator model D. The goal of the discriminator is to learn how to well differentiate real samples from fake samples. That is, D is a differentiable function that tried to obtain a value near 1 when applied to real samples, and a value near 0 when applied to fake samples. The goal of the generator is to become better and better at generating samples looking like real ones, such that the discriminator has a more difficult job to differentiate fake samples from real samples. The input of G is represented by noise, and its output (the generated sample) is then the input for D. Image from Goodfellow (2016).

either gradient penalty (GP) (Gulrajani et al., 2017) or Lipshitz continuity (LP) (Petzka et al., 2017). In this scenario, the discriminator does not provide a direct estimation of whether a sample belongs to the real data or not, but rather acts as a critic, and assists the computation of the Wasserstein distance between the training data and the generated data (Arjovsky et al., 2017). The discriminator provides the generator with a loss that can be trained until optimality (Arjovsky et al., 2017), and gives stability to the GAN and avoids mode collapse[2] (Dong et al., 2018).

[2]Mode collapse is a form of GAN failure. The generator starts to produce the same set of outputs over and over, provoking an *over-optimization* of these outputs. As a consequence, the discriminator

Figure 1.6: **Examples of generated samples.** Left panels: visualization of randomly generated samples after training a GAN (Goodfellow et al., 2014) on MNIST dataset (number from 0 to 9), above, and TFD dataset (black and white faces), below. Right panel: visualization of randomly generated samples after training DCGAN (Radford et al., 2015) on SUN dataset (bedroom pictures). Image from Goodfellow et al. (2014)(left panels) and Goodfellow (2016) (right panel).

GANs' advantages are represented by the computational power and realistic sample production. Figure 1.6 shows random samples generated after training different types of GANs on different datasets. It is possible to observe how the samples belong to the model distributions (Goodfellow et al., 2014). At the same time, GANs could have several problems, coming for example from the dataset, or from the optimization algorithm. If the optimizer is not perfect, underfitting could arise, or, if the training data size is limited, overfitting could arise (Goodfellow, 2016). Moreover, GANs could not reach convergence, due to the high non-convex dimensional space they are dealing with. Indeed, G and D

can't learn how to discriminate the generations. This implies that the generator does not produce a wide enough variety of outputs.

are non-convex parametric functions (e.g., deep neural networks) that do not guarantee convergence. There are two ways in which GANs could fail to converge: 1) *oscillation*, that is the GANs is trained for a very long time and generates many different categories of samples, but never generates better samples; 2) *mode collapse*, that is the generator produces always the same sample, or samples belonging to the same theme (Goodfellow, 2016). Mode collapse is the worst form of non-convergence and the one that happens more often. This comes from the fact that GANs are trained using gradient descent which is leading, at the same time, to convergence (when solving the classic min-max game mentioned above) and to concentrate on most likely points (when solving the reverse max-min problem) (Metz et al., 2016). To avoid mode collapse, a solution is to look at the features of an entire mini-batch of samples when examining a single sample: if that sample is too close to the other samples in the mini-batch, it could be rejected (Salimans et al., 2016).

Evaluation of GANs, that is the capability of the generative model to produce a realistic output, is a challenging and debated topic (Goodfellow, 2016; Theis et al., 2015). Models with good likelihood could produce bad samples, and, vice versa, models that produce good samples could have a bad likelihood. The likelihood itself is not easy to estimate. Also, there is not a unique way to measure how good samples are. Lately, Borji (2019) proposed a combination of a quantitative and a qualitative measure to evaluate GANs' performance. The first allows to understand the capability of the generator of reproducing a wide enough variety of samples, and to evaluate the model in a low-dimensional complexity (i.e., it is usually a value representing the entropy, the variance, etc.). The latter helps the inspection of the generated samples and is usually based on human judgment.

1.3.4 Sound generation models

Several models for speech and music generation have been proposed. Not all of them are GANs, and often the basics to define the architecture of the model have been imported

from image generation studies. Below, the main characteristics of several models are highlighted. In particular, these models are on the one side well-known models for sound and speech generation, on the other side models that helped me understanding the desirable properties of a generative model and how to investigate them.

Oord et al. (2016b) proposed a deep neural network (called PixarCNN) that operates directly on the raw waveform. It has been used to define **WaveNET**: its architecture is based on convolutional neural networks combined with a non-linear function to reconstruct the conditional distributions over the samples. The model shows good results when applied to a dataset of different speakers in the context of speech generation and text-to-speech (i.e. the model is able to generate a realistic speech starting from a dataset of written texts). Moreover, when trained with music, WaveNET shows the ability to generate novel sounds that were not present in the training dataset (Oord et al., 2016a). Such a model enables the generation of realistic sound but does not provide a low-dimensional representation of it. The backside of WaveNET is the computational time: the model takes two minutes to synthesize one second of audio (Goodfellow, 2016). To evaluate the sound generated using Wavenet, Oord et al. (2016b) used blind subjective evaluation.

Magenta from Google AI is an open-source research project devoted to understanding the role of machine learning in the creative process. Engel et al. (2017) developed **NSynth** (Neural Synthesizer), a music synthesizer that uses deep neural networks to generate sounds at the level of individual samples. The architecture is based on a WaveNet-style autoencoder model able to meaningfully represent the sound space. That is, the sound is encoded in a latent space that preserves uniquely the identity of the timbre and the dynamics. This allows both to obtain a lower-dimensional representation of the input space and to explore new sounds by controlling over the two components of the latent space. To qualitatively evaluate NSynth, they trained a multi-task classification network able to predict pitch and label. The obtained accuracy on the NSynth generated data is roughly equal to the original audio. Lately, Roberts et al. (2018) proposed **MusicVAE**, a hierarchical variational autoencoder for learning latent spaces for musical scores. The basic structure of the model was previously proposed for sequential data (Bowman et al.,

2015), and a hierarchical decoder has been introduced as proposed in SketchRNN for hand drawing (Ha and Eck, 2017). The novelty is represented by the introduction of a sequence of RNNs that autoregressively decode the input sequence: each RNN processes a segment of the input of the decoder and output a new sequence, which is processed through a final RNN decoder. The experiments on music data show good performances: on the one hand, samples resembling real ones can be produced; on the other hand, interpolations between produced samples show continuity in the latent space.

Pascual et al. (2017) proposed **SEGAN**, a Speech Enhancement Generative Adversarial Network that works end-to-end with waveform data (that is, it takes waveforms as input and it provides waveform generation). The generator network is composed of two convolutional neural networks: one takes a noisy waveform as input and extracts acoustic features, encoding the input in a low-dimensional vector (i.e., a latent variable); the other takes a latent vector as input, apply convolution, and outputs a waveform having the same dimension of the input waveform. The discriminator learns how to discriminate a generated sound from a real one, and helps the generator to correct its output waveform towards a realistic distribution. The adversarial training has been done using gradient backpropagation. To evaluate the performance of SEGAN, both quantitative and qualitative measures have been used. For instance, Pascual et al. (2017) measure speech distortion and background noise influence (quantitative evaluation), or asked for human judgment on generated samples (qualitative evaluation).

Dong et al. (2018) proposed **MuseGAN**, a music generator trained on a multi-track piano-rolls dataset. They use symbolic timing, that is each beat has the same length, and propose three different music generation models. First, multiple generators work independently and generate music, to receive later critics from several discriminators. Then, a single generator creates a multi-channel (each representing a specific track), and a discriminator differentiates real samples from generated samples. Finally, a hybrid model combines the two previous models to build the music composer: it can use different network architectures and have different inputs). To evaluate MuseGAN, they used a quantitative measure based on the characteristics of the dataset they are using (i.e multi-

track piano rolls): for example, they measured the distance between tones and the drum pattern. They applied the metric both to the training dataset and to a dataset of generated samples: the statistics of the real and the generated data should become closer as the training goes on.

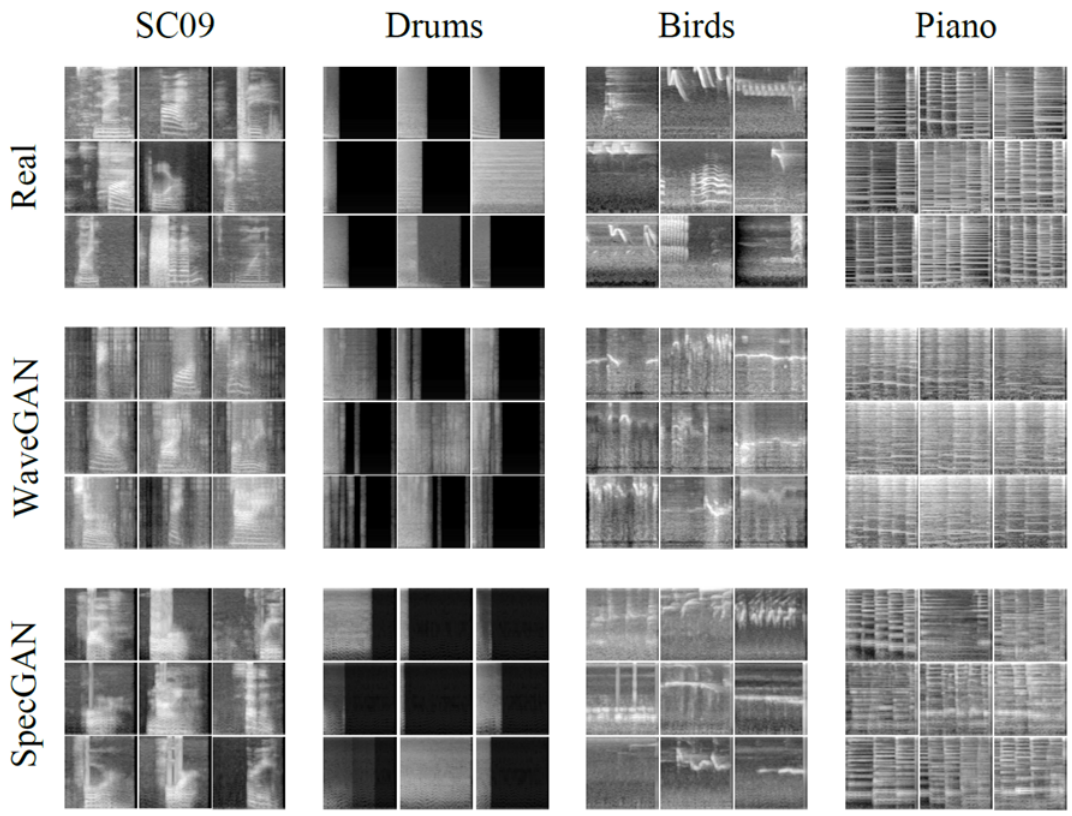


Figure 1.7: **Examples of samples for WaveGAN and SpecGAN models.** Visualization of random samples from the training dataset (top line), the generated data using the WaveGAN generator (middle line), the generated data using the SpecGAN generator (bottom line). Respectively, the first column shows the comparison between the real data made of speech recordings and the generations obtained after training), the second column shows the comparison between the real data made of drums recordings and the generations obtained after training, the third column shows the comparison between the real data made of wild birdsongs recordings and the generations obtained after training, the last column shows the comparison between the real data made of piano recordings and the generations obtained after training. Image adapted from Donahue et al. (2018).

Donahue et al. (2018) proposed two GANs for music, speech and, to a lesser extent, birdsong generation: **SpecGAN** enables spectrogram generation, **WaveGAN** enables waveform generation. Both the models are based on Deep Convolutional GAN architecture (Radford et al., 2015), and eventually, introduce variations in order to adapt the model to the dataset. On the one hand, in SpecGAN the spectrogram representation is designed in order to suit the DCGAN image generation and to allow inversion (which is needed to obtain the waveform after the generation). On the other hand, WaveGAN architecture has been modified in order to take into account waveforms as input, and the WGAN-GP (Gulrajani et al., 2017) strategy has been used to optimize the loss function. Both models can generate samples of speech that are intelligible by humans. Nevertheless, WaveGAN generates sounds of better quality with respect to SpecGAN. This could be due to the sub-optimality of the spectrogram inversion algorithm (here, Griffin-Lim inversion (Griffin and Lim, 1984) has been used). Moreover, the frequency domain produced by WaveGAN resulted to be more coherent with the real data. Good results have been obtained for music generation (drums and piano), speech and birdsong, with a wide variety of generated sounds.

To evaluate WaveGAN performances on a speech dataset, Donahue et al. (2018) used several quantitative measures. First, they computed the Inception Score (IS), a metric proposed previously by Salimans et al. (2016) that represents the capacity of the generator of producing a wide enough diversity of samples. In addition, they computed the Euclidian distance in the space of log-Mel spectrograms (1) within the training and the generated data separately, and (2) between the training and the generated data. In this way, they show the intra-diversity of the generated data and the ability of WaveGAN of producing sounds not belonging to the training dataset. Finally, they used human judgment as a qualitative measure.

1.4 Objectives of the thesis

Vocal learning models can either aim to obtain a vocal learning machine or to understand the biology of vocal learning. The first objective of this thesis is to understand the biology underlying vocal learning (e.g., learning phases, behavioral studies, neurobiological basis). Secondly, starting from the literature about vocal learning models, we aim to define a schema that could help to compare the existent models in terms of their goal, assumptions and components (e.g., learning algorithm, motor control function). Moreover, we aim to understand the computational tools available to implement the components of a vocal learning model (e.g., dynamical systems and generative models to implement the motor control function) to identify which ones could be used in our model. In particular, we want to test how they work on real recordings (in our case, canary data). Finally, we aim to place this thesis in the context of the existing vocal learning literature and provide perspectives for this work.

Chapter 2

Vocal Imitation in Sensorimotor Learning Models

A Comparative Review

Vocal learning models literature is vast, variegate and full of multidisciplinary contents. The chapter attempts to compare existing vocal learning models from studies on different subjects (humans or songbirds). The objective is to find a common scenario to disentangle a model into components that can be compared across models.

Section 4.1 defines the vocal learning model schema identified as a guideline to compare different models. Section 2.2 contains a gentle introduction to the neuroanatomy of the human and songbird brain, and an analysis of the links between biology and the sensorimotor components. Section 2.3 introduces the reviewed models and their main objectives. Sections 2.4, 2.5 and 2.6 contain the description of the components of the baseline models, and how the reviewed models can be decomposed in terms of them. Section 2.7 contains a discussion about how the reviewed models are positioned with respect to the biological framework. Moreover, on the one hand, it underlines directions to define a more and more bio-inspired model, and, on the other hand, it summarizes how the proposed schema can help in the comparison between different models.

A preliminary short review was previously published within the ICDL-Epirob Workshop on Continual Unsupervised Sensorimotor Learning (Sep 2018, Tokyo, Japan) (Pagliarini et al., 2018b) and then extended to the current chapter. " *Vocal Imitation in Sensorimotor Learning Models A Comparative Review* (Pagliarini et al., 2020) has been published within the Journal of Transactions on Cognitive and Developmental Systems, SI: Continual Unsupervised Sensorimotor Learning.

Abstract

Sensorimotor learning represents a challenging problem for natural and artificial systems. Several computational models have been proposed to explain the neural and cognitive mechanisms at play in the brain. In general, these models can be decomposed in three common components: a sensory system, a motor control device and a learning framework. The latter includes the architecture, the learning rule or optimisation method, and the exploration strategy used to guide learning. In this review, we focus on imitative vocal learning, that is exemplified in song learning in birds and speech acquisition in humans. We aim to synthesise, analyse and compare the various models of vocal learning that have been proposed, highlighting their common points and differences. We first introduce the biological context, including the behavioural and physiological hallmarks of vocal learning and sketch the neural circuits involved. Then, we detail the different components of a vocal learning model and how they are implemented in the reviewed models.

Contents

2.1 Introduction

Humans and animals such as songbirds show imitative vocal learning: they are able to produce a motor command that replicates a previously experienced auditory stimulus (Brainard and Doupe, 2002; Heyes, 2001, 2012; Kuhl, 2004). Imitation implies a causal relationship between the observed stimulus and the produced action, and requires a mechanism to translate the sensory input into motor commands (Heyes, 2001). Humans and animals are able to perform complex imitation, this is illustrated by the imitation of novel action sequences in response to environmental cues (Heyes, 2012).

Imitative vocal learning, and more generally sensorimotor learning, are the subject of behavioural, anatomical, physiological and computational studies. Taking into account the biological evidence and constraints revealed by experimental investigations of the underlying brain circuits, many previous studies have attempted to implement imitative learning in computational models. The aim of this review is to identify and compare the various components of existing vocal learning models to provide an integrated and organised view of the literature. While we focus our analysis on vocal learning, the principles addressed here may also apply to sensorimotor learning models in general. To analyse and compare the existing models, we will now define the core components at play in models of vocal learning.

As depicted in Figures 2.1 and 2.2, the representations needed for a minimal vocal learning model can be cast into three spaces (Oudeyer, 2005): motor, sensory and perceptual/internal space. In addition, one needs to define a learning framework and define the connections between the spaces: a motor control function and a sensory response function. The learning framework contains the architecture, the learning algorithm, the evaluation and the exploration strategy (see Table 2.6). We define the input and output spaces of the learning algorithm as the *learning domain* and the *learning image*[1]. The motor space corresponds either to the muscle activation patterns sent to the vocal or-

[1]The idea is to conceptualise the learning algorithm as a mathematical function going from the domain, called *learning domain*, to its co-domain, called *learning image*. For simplicity, we will use *learning image* instead of *learning co-domain*.

gan (e.g. larynx and syrinx for human and birds respectively) or articulatory parameters (e.g. the tongue height) for humans. The sensory space, in the case of vocal learning, represents the physical space of the sound. The perceptual space corresponds to the neural representation of perceived vocalisations in the brain (e.g. acoustic features as pitch in birdsong or first formants in speech). Space representations are implemented as vectors or trajectories (i.e. sequences of vectors) in these multi-dimensional spaces.

Figure 2.1 shows the canonical model including an action-perception loop: the perceptual space is connected to the motor and sensory spaces through a sensory system and a motor control device. Depending on the modeller's choice, the perceptual and motor spaces may be linked through an inverse model, or both an inverse and a forward model (see definition of internal models and in particular inverse and forward model in Section 2.6). The learning domain (i.e. input space of the learning algorithm) is the perceptual space in the case of inverse models, and the motor space in the case of forward models.

An internal representation of the goal could lie in the perceptual space, or alternatively as shown in Figure 2.2, in a separated space if it is encoded independently of the sensory processes of experienced vocalisations. In such a model, an internal representation of the goal is used as the learning domain and hence, it is *non-perceptual*. In the present review, we call *goal-to-motor model* the connections between the internal representation and the motor space. The sensory processing of the produced vocalisations may still be implemented downstream from the motor space, for instance to provide a reward signal that guides learning in a reinforcement learning framework.

Whichever the particular learning framework and mechanisms used, the modelled agent must explore either the goal space or the motor space to later adjust its production. Such a vocal exploration may be purely random or more sophisticated (e.g. intrinsically motivated exploration) (Oudeyer et al., 2007). To explore either the motor space or the goal space, and to improve current vocal performance, learning models rely on the evaluation of the produced vocalisation. The aim of the evaluation is to obtain a measure that defines an error signal and/or a reinforcement signal, later used by the learning

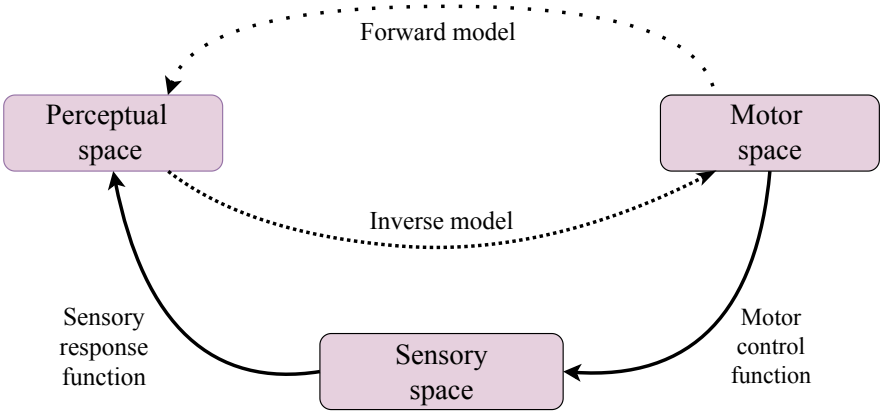


Figure 2.1: **Sensorimotor model with a action-perception loop.** The *motor control function* generates a sensory representation (a sound in the more complete models) given the motor command parameters. This kind of sensorimotor model includes an inverse model, and potentially a forward model. One of the advantages of a forward model is that it can bias the perceptual representation in order to facilitate the inverse model learning towards perceptuo-motor representations (Wolpert and Kawato, 1998).

framework to update the architecture.

Table 2.1 contains all the acronyms used along the review. Section 2.2 contains an introduction to the neuroanatomy of the human and songbird brains. Additionally, it contains an analysis of the links between biology and the sensorimotor components. Section 2.3 details the aim of the reviewed models, giving an overview of the objectives and questions pursued by the modellers. Table 2.2 summarises the aims of the models. Section 2.4 describes the motor control device and its components: the motor space, the articulatory model and the connection with the sensory space. Section 2.5 introduces the representation of the sensory system and its components: the sensory space, the sensory response function and the perceptual space. Table 2.4 contains a summary of the spaces and functions of sensorimotor models. Section 2.6 elaborates on the components

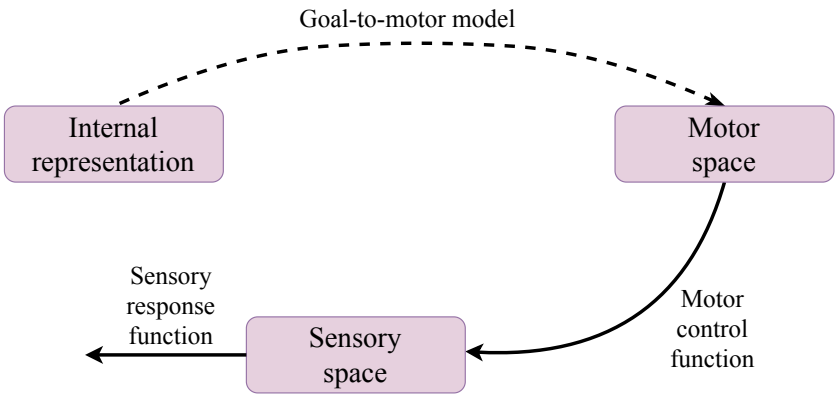


Figure 2.2: **Non-perceptual sensorimotor model.** This kind of model is *non-perceptual* because it has a non-perceptual internal representation of goals. The dotted line represent learned connection from goals to motor commands: we call this the *goal-to-motor model*. Sensory response function processes the sound and can be implemented in various ways depending on the learning framework: it could be used to provide a reward or an evaluation of the learning (for this reason there is an arrow starting from the sensory space, but without a specific output space).

of the learning framework and Table 2.6 summarises the implementations used in the models. Section 2.7 contains a discussion about the reviewed models, their relation with the biological framework introduced in Section 2.2, and further directions are proposed.

2.2 Biological context

We present here the biological context of vocal learning. We first highlight the behavioural phases included in imitative vocal learning in humans and songbirds. Then, the main brain circuits related to song (for birds) and spoken language (for humans) are discussed and compared. Finally, we introduce current mechanistic hypotheses and some biological

Acronym	Extended name
Biological context	
DLM	thalamic nucleus DorsoLateralis anterior par Medialis
aSt	anterior Striatum
aT	anterior Thalamus
HVC	High Vocal Center
LFP	Local Field Potential
LMAN	Lateral Magnocellular nucleus of Anterior Nidopallium
LMC	Laryngeal Motor Cortex
LTD	Long-Term Depression
LTP	Long-Term Potentiation
MNs	Mirron Neurons
RA	Robust nucles of Arcopallium
SMP	Song Motor Pathway
STRF	Spatio-Temporal Receptive Field
Computational Models of the Vocal Tract	
DIVA	Directions Into Velocities of Articulators
ODEs	Ordinary Differential Equations
qTA	quantitative Target Approximation
VTL	VocalTractLab
VLAM	Vocal Linear Articulatory Model
Learning Framework	
COSMO	Communicating Objects through SensoriMotor Operations
CMA-ES	Covariance Matrix Adaptation - Evolution Strategy
ESN	Echo State Network
FF NN	Feed Forward Neural Network
IAC	Intelligent Adaptive Curiosity
O	Optimization algorithm
RBF	Radial Basis Function
RL	Reinforcement Learning
RNN	Recurrent Neural Network
S	Supervised learning
SOM	Self-Organizing Map
U	Unsupervised learning
Algorithms	
BMU	Best Matching Unit
F0	Fundamental frequency
GMM	Gaussian Mixture Models
HPF	High-Pass Filter
LDA	Linear Discriminant Analysis
LPF	Low-Pass Filter
MFCC	Mel-Frequency Cepstral Coefficients
MSE	Mean Square Error
PCA	Principal Component Analysis
SSE	Sum of Squared Error

Table 2.1: Summary of the acronyms used in the review.

constraints that should be taken into account while defining a vocal learning model: mirror neurons' activity and their putative function in vocal learning, experimental evidence for synaptic plasticity and the sensory representation of vocalisations. In the last three subsections, the literature comes mainly from songbirds, but may also serve as biological support for human studies.

2.2.1 Learning phases and behaviour

From a behavioural point of view, speech learning in humans and song acquisition in birds are made up of the same developmental behavioural phases (Doupe and Kuhl, 1999; Kuhl, 2004, 2000). Figures 2.3 and 2.4 show the first year of speech perception (green background) and production (pink background) development in infants (adapted from Kuhl (2004)) and songbirds (adapted from Doupe and Kuhl (1999)). In babies, as shown in Figure 2.3, sensory learning starts immediately after birth and allows the infant to discriminate the phonetic contrasts specific to the learned language. This process, also known as categorical learning, is described in Subsection 2.2.3. Vocal production starts with the production of non-speech sounds, also shortly after birth. After this preliminary phase, sensorimotor learning starts: speech-like sounds are first produced erratically, then "canonical babbling" emerges and the first words are produced by the infant around the age of one year (Kuhl, 2004; Moulin-Frier et al., 2014).

In birds, as shown in Figure 2.4, the sensory learning phase enables juveniles to build a neural representation of adult vocalisations, which would later guide vocal production (Doupe and Kuhl, 1999). Juveniles have a species-specific predisposition and listen to the sounds produced by their parents (Doupe and Kuhl, 1999). Then, during the sensorimotor phase, the young birds start to vocalise, initially producing babbling sounds and then adapting their vocal output to imitate previously heard vocalisations (Brainard and Doupe, 2002). Finally, the produced vocalisation becomes more and more stereotyped and vocal plasticity significantly drops. This final phase, when song production converges towards the stereotyped adult song, is called crystallisation in birds (Doupe and Kuhl,

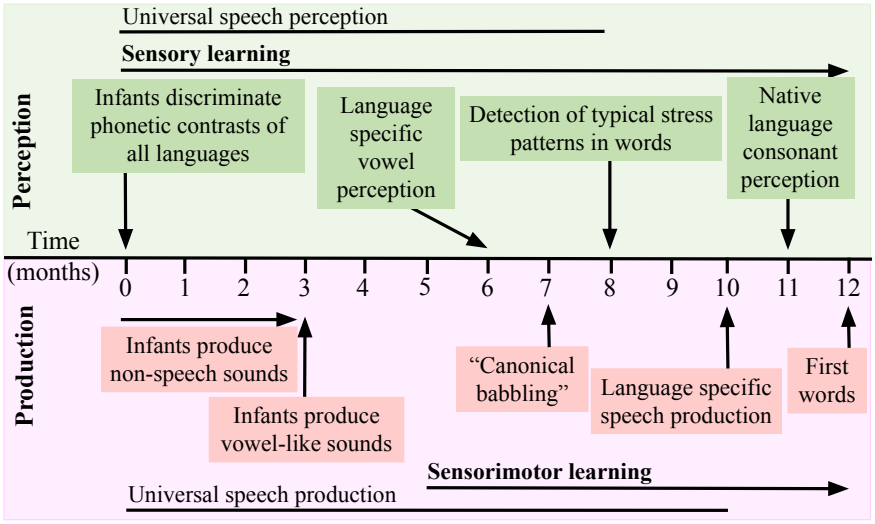


Figure 2.3: **First year of infant speech-perception and speech-production development.** Speech perception development (green background) is characterised by a sensory learning phase that shapes perception, from an initially universal perception to language-specific phoneme discrimination. Speech production development (pink background) is characterised by some preliminary phases followed by sensorimotor learning, where "canonical babbling" takes place. Image adapted from Kuhl (2004).

1999; Kuhl, 2000).

2.2.2 Neuroanatomy of human and bird brain

Figure 2.5 shows the brain pathways controlling song in songbirds (upper panel) and spoken language in humans (lower panel). In both cases, there are two main pathways (Chakraborty and Jarvis, 2015): the posterior vocal motor pathway (plain black arrows) and the anterior vocal learning pathway (plain white arrows). In addition, there are connections between the two pathways (dashed black arrows) and specialised direct projection to vocal motor neurons (plain red arrows).

The vocal production pathway in birds (Figure 2.5(a)) projects from HVC (used as a proper name[2]) to robust nucleus of arcopallium (RA) (plain black arrows, upper panel).

[2]HVC was originally High Vocal Center.

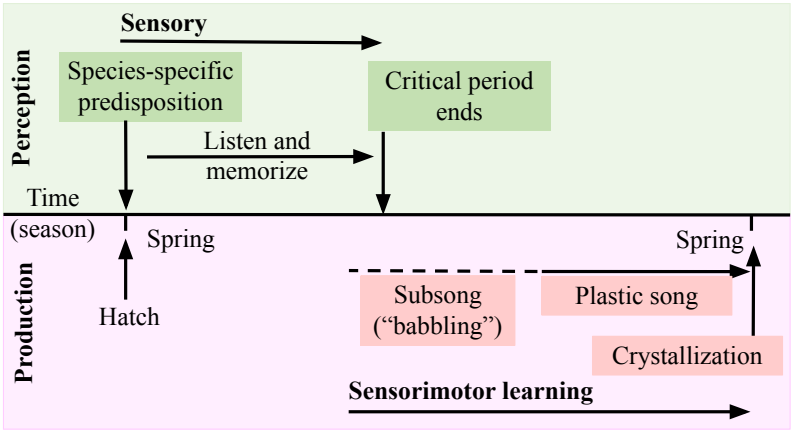


Figure 2.4: **Imitative learning phases in birds.** Three main phases characterise imitative learning in songbirds: the sensory learning phase, the sensorimotor learning phase (starting with subsong and continuing with a plastic song), and crystallization of the song (i.e. convergence to adult song). Image adapted from Doupe and Kuhl (1999).

RA and its analogous in humans, represented by the laryngeal motor cortex (LMC), connect directly to vocal motor neurons (plain red arrows, lower panel) providing the motor output (controlling the larynx in humans or syrinx in birds) (Chakraborty and Jarvis, 2015; Jarvis, 2019). The vocal learning pathway is responsible for vocal imitation and plasticity: it forms a basal ganglia-thalamo-cortical loop. In birds, as shown in Figure 2.5(a), it involves the song-related song nucleus Area X, the thalamic nucleus dorsolateralis anterior pars medialis (DLM, sometimes called aDLM) and the lateral magnocellular nucleus of the anterior nidopallium (LMAN, more generally called MAN). The indirect projection onto RA from Area X, through DLM and LMAN is represented by a dashed black arrow (Chakraborty and Jarvis, 2015; Jarvis, 2019). In humans, in Figure 2.5(b), the vocal learning pathway presumably includes Broca's area (one of the main areas of the language cortex in humans along with Wernicke's area and superior temporal gyrus (Friederici, 2011)), the anterior striatum (aSt) and anterior thalamus (aT).

The neuroanatomical structure of the vocal control circuit in human and bird provides the anatomical basis for bio-inspired models of vocal learning that often question the

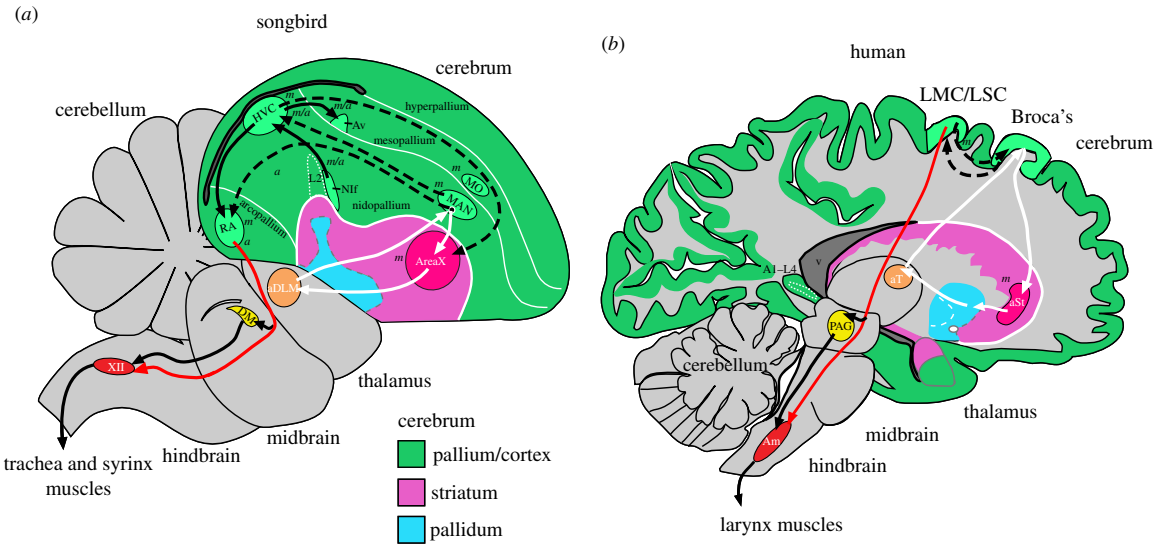


Figure 2.5: **Brain pathways controlling (a) song in songbirds and (b) spoken language in humans.** The posterior vocal motor pathway (plain black arrows) is also called vocal production pathway, since it involves direct projections to motor neurons. The anterior vocal learning pathway (plain white arrows) is responsible for vocal imitation and plasticity. In addition, there are the connections between the two pathways (dashed black arrows) and specialised direct projection to vocal motor neurons (plain red arrows). In panel (a), *a* indicates a region where there is an auditory neural activity, *m* a region where there is motor neural activity, *m/a* a region where both auditory and motor neural activities are present. In panel (b), *v* indicates the ventricle space. Image from Chakraborty and Jarvis (2015), CC-BY 4.0 license.

function of specific brain areas and/or the connections between them. Please refer to Section 2.3 and Table 2.2 for studies making explicit reference to the neuroanatomy of the brain.

2.2.3 Sensory system

The auditory system of mammals and birds builds up selective responses to auditory stimuli. Ultimately, auditory selectivity may give rise to categorical perception, the tendency to perceive a continuous change in sensory space (e.g. sound) as discrete percepts (e.g. phonemes or bird syllables). This ability exists in both humans and birds (Kuhl, 2000, 2004). During infant first months, the acoustic differences detected influence the selection

of phonetic units, and infants become more sensitive to the units that are important for the language they hear (Kuhl, 2000, 2004). Similarly in birds, neural selectivity for the imitated song develops slowly during song ontogeny (Brainard and Doupe, 2002).

In birds, the auditory system is involved in the discrimination of songs, and relies on the temporal cues and pitch of the song to provide information about the identity of the singer (Hahnloser and Kotowicz, 2010). Song-selective responses (with different responses to the bird's own song and other's vocalisations) have been observed in various high sensory brain areas. The sharp auditory selectivity of neurons in these high sensory areas emerge from a multi-stages auditory pathway that starts from the inner ear. At lower stages of this pathway, sensory responses evoked by the playback of songs or other sounds (including white noise) are well modelled using a linear summation of spatio-temporal receptive fields (STRF) (Theunissen et al., 2000). Higher in the auditory pathway, responses become sparser and more non-linear (i.e. less well modellable by such a linear model) (Hahnloser and Kotowicz, 2010).

Interestingly, experimental studies in birds have revealed that some auditory neurons also respond to perturbations of the auditory feedback during singing (Keller and Hahnloser, 2009; Hahnloser and Kotowicz, 2010). This highlights the fact that the whole pathway from high auditory area to motor areas could be involved in the recognition of tutor or conspecific songs and in the evaluation of the bird's auditory feedback.

2.2.4 Mirror neurons and perceptuo-motor coherence

Some neurons, called mirror neurons (MNs) show a similar response during the perception and the production of a motor or vocal gesture (Rizzolatti et al., 1996; Gallese et al., 1996; Oztop et al., 2006; Prather et al., 2008). Convergence of sensory and motor signals in the same neurons points to a possible mechanism to enable vocal learning (Heyes, 2001): during vocal production, auditory feedback could activate a sensory neural population directly connected to motor neurons driving song production, leading to a strengthening of connections between sensory and motor neural populations through Hebbian learn-

ing (Hahnloser and Ganguli, 2013; Tramacere et al., 2019). These connections could be the substrate of internal models (Kawato, 1999; Giret et al., 2014). However, mirror neurons have only been reported in adult songbirds until now, and it remains unclear whether they are selectively responding to the tutor song following the sensory learning phase. Alternatively, song-related auditory responses may emerge only after the end of the sensorimotor learning phase, ruling out a role for these neurons in song acquisition (Tramacere et al., 2019).

In humans, different theories try to explain why there is activation of motor areas during speech perception (Liberman and Mattingly, 1985; Wilson et al., 2004; Fowler, 2016). For example, the Perception-for-Action-Control Theory (PACT) (**?**) highlights how speech percepts are related not only to sounds, but also to motor gestures: speech perception could be biased by articulatory invariant commands.

Syllables are perceptuo-motor by essence: i.e. perception shapes action (e.g. some abstract representation of motor gesture can be recovered to disambiguate perception) and, at the same time, action shapes perception (e.g. motor gestures are "selected for their functional and perceptual value for communication" (**?**)). An example is the fact that acoustic features can change abruptly when changing the jaw height or jaw cycle, producing phase transition in the perceptuo-motor phase space diagram (**?**).

2.2.5 Learning rules and synaptic plasticity

Learning rules implemented in artificial neural networks are often inspired by biological synaptic plasticity. Evidence for synaptic plasticity in the songbirds song-related network has been highlighted recently (Boettiger and Doupe, 2001; Ding and Perkel, 2004; Sizemore and Perkel, 2011; Mehaffey and Doupe, 2015). The various sites of synaptic plasticity could underlie separate learning processes.

Plasticity in the thalamo-cortical synapse of the learning pathway may subserve early sensory learning (Boettiger and Doupe, 2001). Still, in the learning pathway, long-term potentiation (LTP) in Area X is modulated by dopamine (Ding and Perkel, 2004). LTP

provides experimental evidence for a three-factor learning rule as those often used to model reinforcement learning processes in neural circuits (Legenstein et al., 2010). Indeed, dopamine often mediates reinforcement signals (Schultz, 1998), and several vocal learning models borrow concepts and algorithms from reinforcement learning (RL) theory (Goldberg et al., 2013). In this framework, the progressive improvement of vocalisations observable in the behaviour reflects a trial-and-error strategy guided by the internal evaluation of the produced vocalisations (likely through its comparison with previously experienced adult vocalisations) as well as external rewarding cues directly provided by the adults (Doupe and Kuhl, 1999).

Then, the long-term depression (LTD) of RA recurrent collateral synapses (i.e. between projection neurons) is limited to the song learning critical period; this could implement the pruning of unnecessary connections within RA (Sizemore and Perkel, 2011). The connections in the HVC-RA network are thought to be formed by a dense network that provides many paths for the descending motor signals, and only circuits that were active during singing need to be maintained. This is consistent with the high variability of juvenile's song or infant babbling (Sizemore and Perkel, 2011; Darshan et al., 2017).

Finally, recent evidence for synaptic plasticity in the inputs to RA neurons from HVC and LMAN may provide a key element to model the interaction between the motor and learning pathways during learning (Mehaffey and Doupe, 2015). Indeed, naturalistic stimulation patterns drive opposing changes in the strength of RA's inputs from HVC or LMAN. The extrapolated learning rule may allow the transfer of motor corrections initially driven by LMAN inputs and later consolidated in the motor pathway (Andalman and Fee, 2009).

2.3 Aims of the models

The topics of the reviewed models are either speech perception and production development in humans, or song acquisition in birds. The second column of Table 2.4 contains the subject of the study of each paper that is reviewed: either humans (*"H"*) or song-

Sensorimotor integration	Brain area functions	Architecture/ plasticity rule
Bailly (1997) Westerman and Miranda (2002) Moulin-Frier et al. (2015)	Doya and Sejnowski (1998) Troyer and Doupe (2000) Fiete et al. (2007) Cohen and Billard (2018) Barnaud et al. (2019)	Doya and Sejnowski (1998) Troyer and Doupe (2000) Fiete et al. (2007) Howard and Huckvale (2005) Oudeyer (2005) Howard and Messum (2007) Kröger et al. (2009) Liu and Xu (2014) Philippsen et al. (2014) Murakami et al. (2015) Warlaumont and Finnegan (2016) Najnin and Banerjee (2017) Pagliarini et al. (2018a) Barnaud et al. (2019)

Realistic vocal tract	Exploration	Social interactions
Doya and Sejnowski (1998) Howard and Huckvale (2005) Moulin-Frier and Oudeyer (2012) Murakami et al. (2015) Philippsen et al. (2016) Teramoto et al. (2017) Howard and Birkholz (2019)	Moulin-Frier and Oudeyer (2012) Moulin-Frier et al. (2014) Philippsen et al. (2016) Forestier et al. (2017) Acevedo-Valle et al. (2018)	Oudeyer (2005) Lyon et al. (2012) Moulin-Frier et al. (2015) Acevedo-Valle et al. (2018)

Table 2.2: Summary of the main objectives pursued by the authors of the reviewed models.

birds (*"SB"*). In both cases, the focus is on early stages of learning, when babbling takes place (see Section 2.2 for more details about learning phases in humans and songbirds). Beyond the general topic, there are several objectives and questions that the authors have pursued. An overview of these objectives is shown in Table 2.2: (i) investigate the effects of sensorimotor integration on the model definition, (ii) test the biological plausibility of hypotheses for the function of vocal learning brain areas, (iii) test a particular architecture and/or plasticity rule, (iv) include a realistic vocal tract, (v) test different types of exploration, and (vi) model social interactions.

2.3.1 Effects of sensorimotor integration

Some authors aim to study sensorimotor integration and its effect on sensory and motor space representations: Bailly (1997) is interested in sensorimotor redundancy given

the constraints imposed by the articulatory system; Westerman and Miranda (2002) are concerned by the effect of auditory perception and production on the development.

2.3.2 Biological plausibility

Many authors are interested in modelling song-related pathways in birds, and in the study of auditory feedback. Troyer and Doupe (2000) test several hypotheses about anterior vocal learning pathway including HVC-RA connections, efference copy and auditory feedback. Doya and Sejnowski (1998) test the hypothesis that LMAN drives slow exploration in the connection from HVC to RA. Alternatively, Fiete et al. (2007) hypothesise that LMAN produces transient song perturbations by driving rapid conductance fluctuations in RA neurons. In the context of speech production and perception, some authors developed models inspired by functions of brain areas (Kröger et al., 2009). Cohen and Billard (2018) tested the hypothesis that human brain areas are shared in language understanding and production, and their implication in goal-directed actions using active language learning and social babbling. Barnaud et al. (2019) tested the hypothesis of idiosyncrasies (individual specificity) in production and perception; moreover, they tested the inter-individual variability in auditory and motor prototypes within a given language.

2.3.3 Learning architectures and algorithms

Some authors test the hypothesis that the anterior vocal learning pathway works as an actor-critic system and implement a gradient-based reinforcement learning rule. This is the case of Doya and Sejnowski (1998) and Fiete et al. (2007). Reinforcement learning is implemented also by Howard and Messum (2007) and Warlaumont and Finnegan (2016): they test the hypothesis that actions are reinforced based on auditory salience. Finally, Troyer and Doupe (2000) combine Hebbian leanring and a reinforcement learning signal in their architecture. Alternatively, other authors learn internal models using different learning rules to update the synaptic weights matrix representing the connections between motor commands and goal representations. Howard and Huckvale (2005) compare direct

inverse mapping and distal supervised learning in the context of speech generated both by a real human subject and by a synthesizer; Philippsen et al. (2014) aim to understand how to reduce the need for supervised training using only acoustic examples learning efficiently an inverse and a forward model. Oudeyer (2005) and Pagliarini et al. (2018a) test a normalised Hebbian rule to learn the inverse model. Liu and Xu (2014) test if it is possible to develop the acoustic-to-articulatory model by learning inverse kinematics in speech acquisition. More particular cases are presented by the works from Murakami et al. (2015) who test imitation learning using a recurrent neural network, Kröger et al. (2009) who test a self-organised network (SOM), and Barnaud et al. (2019) who test a Bayesian model of speech communication.

2.3.4 A realistic vocal tract model

One of the objectives of many authors is to take into account anatomical and physiological constraints using a realistic model of the vocal tract for the production of the sound. Many authors want to include such a model in their study: Doya and Sejnowski (1998), Howard and Huckvale (2005), Moulin-Frier and Oudeyer (2012), Murakami et al. (2015), Philippsen et al. (2016), Teramoto et al. (2017). In particular, Howard and Birkholz (2019) test two different vocal tract models, with increasing complexity, both in the case of a real human teacher and in the case of an automatic synthesizer of sounds.

2.3.5 Exploration strategies

Several authors test whether or not mechanisms of intrinsically motivated exploration can self-organise early developmental stages of learning: Moulin-Frier et al (Moulin-Frier and Oudeyer, 2012; Moulin-Frier et al., 2014) compare different exploration strategies (random motor exploration, random goal selection and curiosity-driven active goal selection) to drive learning; Forestier and Oudeyer (2017) focus on body babbling coupling self-generation of goals and imitation learning without any assumptions of capabilities for complex sequencing; (Philippsen et al., 2016) test goal-directed exploration of the target

space and assume that there is no need of visual information. On the contrary Murakami et al. (2015) starts and studies the relevance of visual information. Intrinsically motivated exploration is also in the interest of Acevedo-Valle et al. (2018): they formalise a socially reinforced and intrinsically motivated architecture for sensorimotor exploration to study the impact of social reinforcement on pre-linguistic development.

2.3.6 Social and multi-agent interactions

Many authors are interested in the influence of social interactions during early pre-linguistic development: Acevedo-Valle et al. (2018) study the influence of imitation maternal responsiveness; Lyon et al. (2012) embed their learning system in a humanoid robot that interacts in real-time with naive participants. Moulin-Frier et al. (2015) and Oudeyer (2005) study self-organising properties of coupling perception and production within agents and between agents.

2.4 Motor control

The first step in defining motor control is to choose an appropriate model mapping a motor space (i.e. muscle command) onto a sensory space (i.e. sound or acoustic representation). This section provides definitions of motor spaces and motor control functions that have been used in models.

Table 2.3: **Summary table of the spaces and functions of sensorimotor models.** –: Not Available; Arb.: arbitrary; Dim: dimension; DIVA: directions into velocities of articulators; IMS: identical to motor space; H: human; JSRU: joint speech research unit; LDA: linear discriminant analysis; M: marmoset; MFCC: mel frequency cepstral coefficients; PCA: principal component analysis; S: sound; SB: songbird; SP?: Sound Production?; VLAM: vocal linear articulatory model

	Subject	**Motor space**		**Motor control**	**Sensory space: SP(?)**	**Pre-processing of the sound**	**Sensory response**	**Perceptual space/ Internal representation**	
		Dim						Dim	
Bailly	H	8	Lip, larynx, jaw, tongue and apex	Maeda	S	4 Formants (in Hz)	Polynomial interpolation + CDA	2	Discriminant space
Doya and Sejnowski	SB	4	Fundamental frequency and peak frequency of sound, sharpness of band-pass filter, gain of the amplifier	Source-filter model	S	–	–	–	Syllable space (localist encoding)
Troyer and Doupe	SB	40	Coordinates (arb.)	–	–	–	–	40	Syllable space (localist encoding)
Westerman and Miranda	H	29	Interarytenoid, cricothyroid, styloglossus, levator palatini, genioglossus, hyloglossus, mylohyoid, orbicularis oris, masseter	2D ODE model (Pipes' walls + Air pressure)	S	2 Formants (in Hz)	All-zero filter + Autoregr. + Gaussian selectivity	2	Formants (in Hz)
Howard and Huckvale	H	9	Jaw, tongue, lip, voicing, fundamental frequency and larynx height	VLAM	S	Spectrogram via JSRU vocoder	Autocorr.	21 * 30 (t)	Autocorr. estimate for F0 and voicing
Oudeyer	H	3	Lip rounding, tongue height, tongue position.	de Boer model	–	4 Formants (in Barks)	Linear combination + Gaussian selectivity	2 * ? (t)	Acoustic trajectory in a 2D subspace of the formants

Table 2.3: **Summary table of the spaces and functions of sensorimotor models.** –: Not Available; Arb.: arbitrary; Dim: dimension; DIVA: directions into velocities of articulators; IMS: identical to motor space; H: human; JSRU: joint speech research unit; LDA: linear discriminant analysis; M: marmoset; MFCC: mel frequency cepstral coefficients; PCA: principal component analysis; S: sound; SB: songbird; SP?: Sound Production?; VLAM: vocal linear articulatory model

	Subject	**Motor space**		**Motor control**	**Sensory space: SP(?)**	**Pre-processing of the sound**	**Sensory response**	**Perceptual space/ Internal representation**	
		Dim						Dim	
Fiete et al.	SB	12	Pitch period and height + filter linear predictive coeff.	Source-filter model	S	–	–	720 neurons	Neural activity
Howard and Messum	H	4 to 9	Jaw, tongue, lip, voicing, fundamental frequency and larynx height	VLAM	S	Low pass filtered spectrogram	Differenced narrow-band spectrogram	2	Low frequency power + spectral change
Kröger et al.	H	2	Back-front, low-high	Motor plan state	S	3 Formants (in Barks)	Rescale to [0,1]	3	Formants (in Barks)
Howard and Messum	H	10	Jaw, tongue, lip, voicing, fundamental frequency, larynx height and nose	VTCalcs	S	–	–	–	Vector of continuous values
Lyon et al.	H	–	–	eSpeak	S	Phonemes (CMU alphabet)	SAPI 5.4	4 sec.	Phoneme stream
Moulin-Frier and Oudeyer	H	7	Jaw, tongue, lip, separation and larynx height	VLAM	S	3 Formants (in Hz)	Linear combination	2	Subspace of the formants
Moulin-Frier et al.	H	7	7 param. from the PCA on the vocal tract shape	DIVA	S	2 Formants (in Hz)	Rescale to [-1,1]	3 * 2(t)	2 Formants (in Hz) + intensity
Moulin-Frier et al.	H	3	Lip, tongue body and dorsum	VLAM	–	3 Formants (in Barks)	–	3	Formants (in Barks)
Liu and Xu	H	3	Target slope and height, rate of target approximation	quantitative Target Approximation	–	F0 continuous traj. (2-dim)	Sampling	5	Syllable space (Time-normalized F0 samples)

Table 2.3: **Summary table of the spaces and functions of sensorimotor models.** –: Not Available; Arb.: arbitrary; Dim: dimension; DIVA: directions into velocities of articulators; IMS: identical to motor space; H: human; JSRU: joint speech research unit; LDA: linear discriminant analysis; M: marmoset; MFCC: mel frequency cepstral coefficients; PCA: principal component analysis; S: sound; SB: songbird; SP?: Sound Production?; VLAM: vocal linear articulatory model

	Subject	**Motor space**		**Motor control**	**Sensory space: SP(?)**	**Pre-processing of the sound**	**Sensory response**	**Perceptual space/ Internal representation**	
		Dim						Dim	
Philippsen et al.	H	26	22 vocal tract arb. param. + glottis param.	VocalTractLab	S	–	Logaritmic energy + 12 MFCC features	39 * ? (t)	Acoustic trajectory
Murakami et al.	H	20	Tongue, lip, hyoid, jaw, velic, velum shape	VocalTractLab	S	Dual Resonance Non-Linear filter model	Reservoir (1000 units)	4 (+1 empty)	Phoneme classes
Warlaumont and Finnegan	H	2	Jaw and lip trajectory	Praat	S	–	–	–	–
Philippsen et al.	H	24	20 vocal tract arb. param. + glottis param.	VocalTractLab	S	3 Formants + 13 MFCC features	PCA + LDA (10 to 2 dim.)	2	Goal vowel embedding
Forestier et al.	H	7	7 param. from the PCA on the vocal tract shape	DIVA	S	–	DIVA	2 * 5(t)	Acoustic trajectory in the 2D space of the first two formants
Najnin and Banerjee	H	11	11 param. for vocal tract and 2 param. for phonation	DIVA	S	–	DIVA	4 / 12	Acoustic trajectory in the 2D space of the first three formants and phonation/normalized MFCCs
Teramoto et al.	M	3	Air pression, vocal fold tension, time constant	Source-filter model	S	–	–	–	–
Acevedo-Valle et al.	H	13*2	10 position of the articulators + 3 phonation parameters	DIVA	S	2 Formants (in Hz)	Average of trajectories	3 * 2(t)	2 Formants (in Hz) + intonation
Cohen and Billard	H	3	Arb.	–	–	IMS	–	–	Specific need (ex: thirst, hunger)

Table 2.3: **Summary table of the spaces and functions of sensorimotor models.** –: Not Available; Arb.: arbitrary; Dim: dimension; DIVA: directions into velocities of articulators; IMS: identical to motor space; H: human; JSRU: joint speech research unit; LDA: linear discriminant analysis; M: marmoset; MFCC: mel frequency cepstral coefficients; PCA: principal component analysis; S: sound; SB: songbird; SP?: Sound Production?; VLAM: vocal linear articulatory model

	Subject	**Motor space**		**Motor control**	**Sensory space: SP(?)**	**Pre-processing of the sound**	**Sensory response**	**Perceptual space/ Internal representation**	
		Dim						Dim	
Pagliarini et al.	SB	3	Arb.	–	–	IMS	Gaussian selectivity	3	Syllable space (localist encoding)
Howard and Birkholz	H	7	Palate, larynx, pharynx, jaw, lips, teeth, togue	VocalTractLab	S	–	–	–	Vector of continuous values
Barnaud et al.	H	3	Lip, tongue body and dorsum	VLAM	–	2 Formants (in Barks)	–	2	Formants (in Barks)

2.4.1 Motor space

The motor space is used to describe motor articulations parameters (ideally as a function of time). These parameters control the dynamics of vocal tract muscles and glottis (for human control models). A high number of parameters is usually provided but often several can be kept constant, either because they do not have much influence on the sound produced or in order to reduce the number of parameters. The dimension of the motor space depends on the motor control function applied and also on the choices made by the modellers. There is a large variability in the number of dimensions of the motor space: from a low dimensional motor space, which only considers the parameters related to lip and tongue, to high dimensional motor spaces which include almost all the available parameters for the vocal tract and, in addition, the glottis parameters.

2.4.2 Motor control function

In humans, vocal motor control involves the respiratory system, the vocal organs (e.g. tongue, lips, jaw, larynx) and the vocal tract. Although some studies have been conducted in the context of vocalisations, neural mechanisms underlying the diversity of respiratory rhythms are largely unknown (Scharff and Nottebohm, 1991).

A basic model of speech production, therefore, includes a sound source (vocal folds) and a linear acoustic filter (vocal tract) (Fant, 2012). The sound source is the combination of vocal folds vibration output and noise. Such noise can be due to pressure fluctuations or by activities of other parts of the apparatus (e.g. the glottis). Lumped-element models are a class of self-oscillating biomechanical vocal folds models: these low-dimensional vocal fold models couple airflow and biomechanics (Birkholz, 2011). Such low-dimensional models can reproduce characteristics of real vocal fold oscillations (Ishizaka and Flanagan, 1972) and have been largely applied in speech research (Erath et al., 2013): the parameters and the structure of lumped-element models can be tuned to sustained vowel simulations to obtain different frequencies. Additionally, it can be tuned to simulate various vocal registers, e.g. to generate a sequence of sounds (to simulate running speech),

or to study some pathological phonation conditions (e.g. incomplete glottal closure).

Downstream from the sound source, the vocal tract acts as a resonator, filtering the sound as it travels to the outside world. It modifies the original sound wave and changes the balance between its frequency components. The resonance frequencies of the vocal tract are called formants (Ladefoged, 1996). The human vocal tract has been often modelled as a structure of pipes: in the literature, Ordinary Differential Equations (ODEs) models describe air pressure dynamics in the vocal tract. For example, Westerman and Miranda (2002) used a synthesizer which models the vocal system as a structure of pipes, each one having four walls represented as mass spring damper models (useful to model non-linearities): the 2D model equations describe the pipe wall physical behaviour, evolution of the movements and air pressure. Similarly, De Boer model (De Boer, 2000, 2001) has been used by Oudeyer (2005): the synthesizer is based on the interpolation between the formant frequencies of vowels generated by Maeda's articulatory model (Maeda, 1989).

Articulatory synthesizers are based on the same idea and control the vocal articulators: (i) *Praat*, a software for speech analysis containing an articulatory synthesizer, developed by Boersma et al. (1998), and used by Westerman and Miranda (2002), and by Warlaumont and Finnegan (2016); (ii) *VocalTractLab* (VTL), developed by Birkholz et al. (Birkholz, Accessed Sept. 2019; Birkholz et al., 2006), and used by Philippsen et al. (2014, 2016), Murakami et al. (2015), Howard and Birkholz (2019), as well as in speech signal filtering (Gudhnason et al., 2015) or articulatory synthesizer training (Prom-on et al., 2013); (iii) *Vocal Linear Articulatory Model* (VLAM) (Maeda, 1990)) has been used by Howard and Huckvale (2005), Moulin-Frier and Oudeyer (2012); Moulin-Frier et al. (2015), Howard and Messum (2007); (iv) *Directions Into Velocities of Articulators* (DIVA) (Guenther et al., 2006a; Tourville and Guenther, 2011) has been used by Bailly (1997), Moulin-Frier et al. (2014), Forestier et al. (2017) and Acevedo-Valle et al. (2018); (v) *VTCalcs* software proposed by Maeda (Maeda) has been used by Howard and Messum (2011).

Taking inspiration from previously developed vocal tract models, Kröger et al. (2009) proposed to define two parameters (*back-front* and *low-high*) describing the state of the

motor plan and covering the whole articulatory vowel space. Other motor parameters like tongue position and lip parameters are expressed in function of these two motor plan parameters. Two particular cases are given by Lyon et al. (2012) which used eSpeak, a synthesizer that uses a formant synthesizer method (Foundation, 2007) and Liu and Xu (2014), where qTA (quantitative Target Approximation) has been used to mimic the motor control dynamics, controlling them via three parameters related to the target properties.

For modelling song production in birds, an interactive model where nonlinear interaction between timescales enables motor instructions has been proposed. More recently, Alonso et al. (2015) developed a simple time continuous additive neural network model that drives the dynamics of respiratory activity: respiratory patterns can be reproduced and predictions on the timing of HVc activity during the production can be performed. Anatomical properties and small size of birds make the investigation of vocal fold mechanisms difficult. It has been shown that the brain seems unable to control each motor parameter independently but it uses a complex gesture-dependent control scheme to drive the vocal output (Elemans et al., 2015; Srivastava et al., 2015). Different studies have been looking at the properties of vocal motor control in correlation with acoustic features, such as 3D imaging techniques to investigate the control of sound pitch (Düring et al., 2017) or neural recordings analysis to investigate the variations in the song (Sober et al., 2008).

Amador et al. (2013), Doya and Sejnowski (1998), Fiete et al. (2007) model the vocal tract dynamics in birds using ODEs. They include time-dependent constants related to air pressure and syringeal labial tension. The output is the pressure needed to generate the sound. Such a dynamical system is able to synthesise realistic vocalisation sounds if a series of instruction derived from a recorded song input is given (Boari et al., 2015). The model from Amador et al. (2013) has been used by Teramoto et al. with marmoset (Teramoto et al., 2017) in a vocal development study.

2.4.3 Sound production

More realistic models generate sound production through the motor control device: Bailly (1997), Doya and Sejnowski (1998), Westerman and Miranda (2002), Howard and Huckvale (2005), Fiete et al. (2007), Howard and Messum (2011, 2007), Howard and Birkholz (2019), Lyon et al. (2012), Moulin-Frier et al. (2014); Moulin-Frier and Oudeyer (2012), Forestier and Oudeyer (2017), Philippsen et al. (2014), Philippsen et al. (2016), Murakami et al. (2015), Acevedo-Valle et al. (2018), Warlaumont and Finnegan (2016). Some models rather rely on an abstract representation of the vocal output including a discrete set of features (e.g. formants) as in the works from Troyer and Doupe (2000), Oudeyer (2005), Moulin-Frier et al. (2015) and Barnaud et al. (2019).

2.5 Sensory system

The sensory system processes sensory stimuli and leads to a perceptual representation of those stimuli (in the perceptual space). While sensory stimuli may arise from other subjects (e.g. adult vocalisations to be memorised during the sensory learning phase), the production of vocalisations by the motor control apparatus also leads to the stimulation of the sensory system. As mentioned in Section 2.2.3, it provides a feedback of the motor command that allows to compare the perceived vocal production with previously experienced adult vocalisations (e.g. the memorised tutor song in the case of birds). The evoked sensory responses may also be conveyed to the reinforcement system, where an evaluation of the produced sounds leads to a reward signal. The sensory response function is often modelled as a minimal extraction of a low dimensional feature-based description of sounds in vocal learning models.

This section motivates the choice of the sensory response function, the sensory space and the perceptual space. In Table 2.4 we separate the physical space of the sound, which is the sensory space, and its abstract representation, which is composed of a pre-processed sound and the perceptual space. This allows to highlight whether a model has

sound production or not, and to compare them in both cases.

2.5.1 Sensory space

The sensory space is the output space of the motor control device. As mentioned in Section 2.4.3 not all the models include sound production. The most simplistic models do not define the motor control device, leading to a coincidence between motor and sensory space. For instance, this approach has been used by Cohen and Billard (2018) and Pagliarini et al. (2018a) (i.e. Identical to Motor Space (IMS) in Table 2.4).

2.5.2 Sensory response function

The sensory response function acts on the sensory space and drives the activity in the perceptual space, where the auditory stimuli are represented with a lower dimension. This process is the result of one or more steps that lead to a filtered, normalised and/or reduced subspace representing the auditory stimulus. The output is an abstract representation of the sound which represents its encoding in the brain. To highlight the fact that the auditory process is in general not a single-step process, a column (*Pre-processing of the sound*) represents an intermediate step between the real sound and the perceptual space. Most models describe first the sound as a trajectory in the formants' space, varying the dimension of the space (usually from 2 to 4) and the measure unit (either Barks or Hertz). Alternatively, other common examples of preprocessed sound are given by a low pass filtered version of the spectrogram, or the trajectory of the fundamental frequency.

A filter on the spectrogram or on the formant space has been applied by Westerman and Miranda (2002). A linear combination has been used in the works from Oudeyer (2005) and Moulin-Frier and Oudeyer (2012). Furthermore, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) have been applied by Philippsen et al. (2016). An average over sound trajectories has been used by Acevedo-Valle et al. (2018). Alternatively, some authors extracted different features from the sound to build its representation in the perceptual space, or its internal representation. This is the case

for Howard and Messum (2011, 2007), Philippsen et al. (2014), Philippsen et al. (2016), Liu and Xu (2014). Howard and Huckvale (2005) estimate the autocorrelation of the fundamental frequency of the sound and of the voicing parameter (from the motor control).

Nonlinearity in the sensory response function can be introduced defining the auditory activity as a bell-shaped function around the target motor pattern. For instance, this choice has been made in the works of Westerman and Miranda (2002), Oudeyer (2005) and Pagliarini et al. (2018a). A few particular cases are given by the work from Lyon et al. Lyon et al. (2012) where a specific software, called *SAPI 5.4* (Yildiz and Kiebel), has been used to encode the stimulus and by the works of Murakami et al. (2015) where a phoneme representation of the stimulus is obtained from a Random Recurrent Neural Network (RNN), called a *reservoir*.

2.5.3 Perceptual space/Internal representation

The output of the sensory response function is a lower dimensional representation of the sound produced by the vocal apparatus. In the context of humans the sound have been represented in the space of the first 2, 3 or 4 formants (in Hertz or Bark scale) by Westerman and Miranda (2002), Oudeyer (2005), Kröger et al. (2009), Moulin-Frier et al. (2014); Moulin-Frier and Oudeyer (2012); Moulin-Frier et al. (2015), Forestier and Oudeyer (2017), Najnin and Banerjee (2017), Philippsen et al. (2016) and Barnaud et al. (2019), Acevedo-Valle et al. (2018). The latter also consider the intonation as third acoustic parameter. Alternatively, Pagliarini et al. (2018a) propose a localist encoding for the syllables.

The percepts can be given by the spectral properties of the sound: for instance the frequency powers, the power change, the fundamental frequency, the Mel-Frequency Cepstral Coefficients (MFCC), or also pitch and amplitude. This choice has been made by Howard and Messum (2011, 2007), Najnin and Banerjee (2017), Philippsen et al. (2014), Philippsen et al. (2016), Liu and Xu (2014) and Fiete et al. (2007). Alternatively, the percepts can be the classes of phonemes, as in the works by Lyon et al. (2012) and Murakami

et al. (2015).

For any species, it is likely that the representation of a given sound in the perceptual space keeps changing during development, thus making the learning of inverse model even more difficult until the moment when the perceptual space "converged". That is why the vast majority of the models have a sensory response function that does not change during learning and is kept fixed. This can be justified based on the assumption that learning the inverse model only starts at the end of the "universal sensory period", which is the case for some species of birds like sparrows.

Some studies do not have a sensory response function that leads to a perceptual space. These are models with a non-perceptual internal representation of the goals, such as the general model shown in Figure 2.2. As for the perceptual space, the choices made by the author can be found in the last column of Table 2.4. This is the case for the reinforcement learning models proposed by Doya and Sejnowski (2000), Troyer and Doupe (2000), Fiete et al. (2007), Warlaumont and Finnegan (2016), Cohen and Billard (2018) and Howard and Birkholz (2019). In the context of songbirds, a typical choice is to use an arbitrary syllable space given by a localist encoding, as in the works from Doya and Sejnowski (1998) and Troyer and Doupe (2000). Alternatively, Fiete et al. (2007) use the neural activity of a spiking neural network. Finally, in the work from Cohen and Billard (2018) goals are specific needs of the agent (e.g. thirsty, hunger).

2.6 Learning framework

This section introduces the different types of architectures, the learning domains (i.e. perceptual, motor or goal spaces), the learning rules or optimisation algorithms, the exploration strategies that could drive learning, and finally the evaluation measures. Table 2.6 summarises how the reviewed models implement the learning framework.

Table 2.4: **Summary table of the learning frameworks of sensorimotor models.** –: Not Available; BMU: Best Matchin Unit; CMA-ES: covariance matrix adaptation - evolution strategy; COSMO: communicating objects through sensorimotor operations; ESN: Echo State Network; F: forward model; FF NN:feedforward neural network; GMM: gaussian mixture models; H: human; I:inverse model; ilGMM: incremental learning GMM; Int: Internal representation; MSE: mean square error; O: optimization algorithm; RL: reinforcement learning; RBF: radial basis function; RNN: recurrent neural network; S: supervised; SB: songbird; SSE: sum of squared error; SOM: self-organizing maps; STDP:spike timing dependent plasticity; U: unsupervised; X: distribution not specified

	Subject	**Architecture**				**Exploration**			**Learning**	**Evaluation**	
		Internal model (if present)	RNN	**FF NN**	**Other**	**Goal-directed**	**Random**	Dimension		Measure	Space
Bailly	H	I + F		1-layer perceptron			X	Motor	Gradient inversion + deviation measure	Deviation	Real forward measure and its estimation by the interpolator
Doya and Sejnowski	SB			2-layer perceptron with 4 subnetworks			Dynamic perturbation	Motor	Reinforcement learning via stochastic gradient ascent	Correlation	Gaussian filter + normalization of the spectrogram
Troyer and Doupe	SB			2-layer perceptron			X	Motor	Reinforcement signal + Hebbian rule	Correlation coefficient	Matrices of co-fluctuations in activity over syllable epochs
Westerman and Miranda	H	I + F		1-layer perceptron			X	Motor	Hebbian Covariance rule	–	–
Howard and Huckvale	H	I + F		2-layer perceptron			X	Motor	Back-propagation + gradient descent	Similarity	Spectrogram of the sound
Oudeyer	H	F		1-layer perceptron			Uniform	Motor	Hebbian Correlation rule	–	–
Fiete et al.	SB			2-layer perceptron			Dynamic perturbation	Motor	Reinforcement learning via stochastic gradient ascent	MSE	Delayed estimate of performance (pitch and amplitude)

Table 2.4: **Summary table of the learning frameworks of sensorimotor models.** –: Not Available; BMU: Best Matchin Unit; CMA-ES: covariance matrix adaptation - evolution strategy; COSMO: communicating objects through sensorimotor operations; ESN: Echo State Network; F: forward model; FF NN:feedforward neural network; GMM: gaussian mixture models; H: human; I:inverse model; ilGMM: incremental learning GMM; Int: Internal representation; MSE: mean square error; O: optimization algorithm; RL: reinforcement learning; RBF: radial basis function; RNN: recurrent neural network; S: supervised; SB: songbird; SSE: sum of squared error; SOM: self-organizing maps; STDP:spike timing dependent plasticity; U: unsupervised; X: distribution not specified

	Subject	**Architecture**				**Exploration**			**Learning**	**Evaluation**	
		Internal model (if present)	RNN	**FF NN**	**Other**	**Goal-directed**	**Random**	Dimension		Measure	Space
Howard and Messum	H		–	–	–		X	Motor	Reinforcement learning via gradient descent	Auditory salience + effort (voicing degree in VLAM)	Spectral properties of the sound, motor properties
Kröger et al.	H	I + F			SOM		BMU	Motor	Hebbian normalized rule	Distance	Motor pattern estimation
Howard and Messum	H				Optimization	X		Sensory, Motor	Quasi-Newton gradient ascent	Auditory salience, diversity and effort	Spectral properties of the sound, motor properties
Lyon et al.	H		–	–	–	Syllable probability		Perceptual	–	F-measure	Perceptual
Moulin-Frier and Oudeyer	H	I + F			Optimization	Competence progress		Goal	Reaching algorithm	Competence progress	Goal
Moulin-Frier et al.	H	I			Bayesian	Competence progress		Goal	GMM over motor variables	Distance	Perceptual
Moulin-Frier et al.	H	I + F			Bayesian		X	Motor	COSMO	Dispersion Theory formula	Perceptual
Liu and Xu	H	I		2-layer perceptron		X		Goal	Online learning, Backpropagation.	SSE	Perceptual
Philippsen et al.	H	I + F	ESN (firing rate reservoir)				Uniform	Goal	Linear regression	MSE	Perceptual

Table 2.4: **Summary table of the learning frameworks of sensorimotor models.** –: Not Available; BMU: Best Matchin Unit; CMA-ES: covariance matrix adaptation - evolution strategy; COSMO: communicating objects through sensorimotor operations; ESN: Echo State Network; F: forward model; FF NN:feedforward neural network; GMM: gaussian mixture models; H: human; I:inverse model; ilGMM: incremental learning GMM; Int: Internal representation; MSE: mean square error; O: optimization algorithm; RL: reinforcement learning; RBF: radial basis function; RNN: recurrent neural network; S: supervised; SB: songbird; SSE: sum of squared error; SOM: self-organizing maps; STDP:spike timing dependent plasticity; U: unsupervised; X: distribution not specified

	Subject	**Architecture**				**Exploration**			**Learning**	**Evaluation**	
		Internal model (if present)	RNN	**FF NN**	**Other**	**Goal-directed**	**Random**	Dimension		Measure	Space
Murakami et al.	H				Optimization	Confidence levels		Goal	CMA-ES	Confidence levels	Goal
Warlaumont and Finnegan	H		Spiking Reservoir				X	Motor	Reinforcement learning via reward-modulated STDP	Auditory salience	Perceptual
Philippsen et al.	H	I + F		2-layer RBF		Dynamic perturbation + GMM		Goal	Distance in goal space + auditory salience	Competence	Goal
Forestier et al.	H				Optimization	Random + Exploration noise		Goal	Reaching algorithm	Distance	Perceptual
Najnin and Banerjee	H		3-layered RNN		Optimization		X	Goal	Autoencoder and actor-critic network	Distance	Perceptual
Acevedo-Valle et al.	H		–	–	–	Interest model		Goal	ilGMM	Distance	Perceptual
Cohen and Billard	H			1-layer perceptron		Caregiver choice		Goal	Maximization of the reward	Moving average of the reward	–
Pagliarini et al.	SB	I		1-layer percepron			Uniform	Motor	Hebbian normalized rule	Distance	Perceptual

Table 2.4: **Summary table of the learning frameworks of sensorimotor models.** –: Not Available; BMU: Best Matchin Unit; CMA-ES: covariance matrix adaptation - evolution strategy; COSMO: communicating objects through sensorimotor operations; ESN: Echo State Network; F: forward model; FF NN:feedforward neural network; GMM: gaussian mixture models; H: human; I:inverse model; ilGMM: incremental learning GMM; Int: Internal representation; MSE: mean square error; O: optimization algorithm; RL: reinforcement learning; RBF: radial basis function; RNN: recurrent neural network; S: supervised; SB: songbird; SSE: sum of squared error; SOM: self-organizing maps; STDP:spike timing dependent plasticity; U: unsupervised; X: distribution not specified

	Subject	**Architecture**				**Exploration**			**Learning**	**Evaluation**	
		Internal model (if present)	RNN	**FF NN**	**Other**	**Goal-directed**	**Random**	Dimension		Measure	Space
Howard and Birkholz	H		–	–	–		X	Motor	Reinforcement learning via gradient descent	Auditory salience, diversity	Spectral properties of the sound, motor properties
Barnaud et al.	H				Bayesian	Accomodation strategy		Perceptual	COSMO	Dispersion Theory formula	Perceptual

2.6.1 Architecture

The architecture linking the learning domain to the learning image varies between models and different architectures can be used. Biological hypotheses made in a particular model are important to understand the choice of the architecture and the learning rule. For more details about the biological hypothesis refer to 2.2.4 and 2.2.5).

Inverse and forward models (i.e. internal models) are both predictor models: they provide a bi-directional link between the perceptual space and the motor space, when both a forward and an inverse model are included. Inverse models have the aim to provide an appropriate motor command for a given perceptual goal, which is driven by the sensory response; the learning domain is defined by the perceptual space. Forward models describe a causal relationship between motor commands and their corresponding perceptual representations; the learning domain is defined by the motor space. As mentioned in Section 4.1, sensorimotor integration leads to redundancy: this is a fundamental problem with inverse models since introducing such kind of model leads to non-convex problems (Reinhart, 2011). This problem can be approached using the combination of an inverse and a forward model (Jordan and Rumelhart, 1992; Wolpert and Kawato, 1998). Indeed, forward modelling can be used to facilitate the estimation of the current state enabling the learner to modify its action and match the prediction: this switches action and perception representations, and explain the effects of perception on action (Pickering and Garrod, 2013). Else, the non-convexity problem can be solved using a combination of an inverse model and a feedback controller (Kawato, 1990) or "goal-babbling" to drive learning (Rolf et al., 2010).

Alternatively, as shown in Figure 2.2 other models define a non-perceptual internal representation of goals and learn not the connections between the perceptual and the motor space, but the link between the internal representation and the motor space. These models include a sensory space if there is a motor control producing a real sound as output, and a sensory response function, which is used to process the sound and build a reward or an evaluation of the learning.

The structure of the network varies among the models. Some approaches involve feed-forward neural networks (FF NN) in the learning architecture. For instance, a 1-layer perceptron has been used in the works by Pagliarini et al. (2018a), Westerman and Miranda (2002), Oudeyer (2005) and Cohen and Billard (2018). A multi-layer perceptron has been used in the works by Doya and Sejnowski (2000), Troyer and Doupe (2000), Howard and Huckvale (2005) and Liu and Xu (2014). A Radial Basis Function (RBF) network has been used by Philippsen et al. (2016). Alternatively, a reservoir has been used by some authors: Warlaumont and Finnegan (2016) uses a reservoir as a kind of biological implementation of reinforcement learning for high-level control of sequential motor production; Philippsen et al. (2014) use two reservoirs to learn both the forward and the inverse models that link motor space with perceptual space; Najnin and Banerjee (2017) use a 3-layered RNN to define the predictive model that uses a generative network to predict the proprioceptive sensory (representing the perceptual dimension) from a causal state (representing the motor dimension). A Bayesian architecture has been proposed in the works from Moulin-Frier et al. (2014), Moulin-Frier et al. (2015) and Barnaud et al. (2019). Finally, three Self-Organizing Maps (SOM) have been used in the work by Kröger et al. (2009).

2.6.2 Learning domain

The learning domain, in the case of inverse models, coincides with the perceptual space, which contains the representation of the stimuli (how the brain encodes sensory stimuli) and is obtained through the sensory response. In the case of forward models the perceptual space coincides with the output, and the learning domain coincides with the motor space. Alternatively, the learning domain is defined as the internal representation of goals and the sensory response drives a reward that modulates the learning rule. In the latter case, the learning domain may be an abstract representation of the goal, that could represent for instance a sound trajectory. Of course, the internal representation could be considered as a component and not the domain of the learning framework, but we prefer to consider

it as the domain of the learning mechanism (or architecture), in order to know what is needed for the learning or optimisation to be available. The full learning domain, or a sub-part, could be also called goal space in models using goal-driven exploration.

The learning domain might encode a whole song (that is a sequence of syllables) or a single syllable, depending on the choices and aims of the model. For instance, the learning domain can be defined as a syllable space that might encode features, as in the works from Troyer and Doupe (2000), Liu and Xu (2014). Or again, using localist encoding (i.e. one-hot encoding)[3] as in the work from Pagliarini et al. (2018a).

Some authors use sound features to describe the perception of a stimulus, for example intensity in the work from Moulin-Frier et al. (2014), fundamental frequency in the work from Howard and Huckvale (2005), Frequency power and spectral change in the works from Howard and Messum (2011, 2007) or pitch and amplitude in the work from Fiete et al. (2007). Alternatively, the learning domain has been defined as a subspace of the formants in the works from Oudeyer (2005), Moulin-Frier et al. (2014, 2015), Kröger et al. (2009), Acevedo-Valle et al. (2018), Forestier et al. (2017) and Barnaud et al. (2019). Philippsen et al. (2016) define the learning domain as the first two dimensions of the embedding space. Here the stimulus has been modelled using the sound trajectory lying in the correspondent space.

Finally, the learning domain can be identified with the output of specific neural networks architecture or particular software. For instance, in the work from Lyon et al. (2012) the goal is represented by a phoneme stream computed using the software SAPI 5.4 (Yildiz and Kiebel), and the work from Murakami et al. (2015) where an intrinsic learning is defined to build the goal space. In the latter, an Echo State Network (ESN, a specific Recurrent Neural Network (RNN), called reservoir) has been used to learn in advance the goals, and an auditory memory encodes the knowledge of each goal (called auditory memory function).

[3]Localist and one-hot encoding is probably the simplest orthogonal representation one can have. It consists of a binary encoding where an input is represented by one feature at 1 and all the other features at 0: e.g. 4-dimensional vectors [0 1 0 0] and [0 0 0 1] could represent two different inputs with localist encoding.

2.6.3 Learning rule

Different types of learning have been used to model sensorimotor learning: supervised and unsupervised learning, and reinforcement learning. In a few models, an optimization algorithm (instead of a learning rule) has been used to improve motor production.

Unsupervised learning

Biologically, as seen in Section 2.2.5, specificity, cooperativity and associativity are expressed in the neural activity. Computationally, this can be modelled using associative learning rules, which are usually used for building internal models (inverse or forward) and are unsupervised. Hebbian-inspired learning algorithms typically implement associative learning and shape the excitatory links between perceptual and motor representations (Heyes, 2001).

A theoretical inverse model has been proposed by Hahnloser and Ganguli (2013), where an Hebbian-inspired learning rule drives learning. Hebbian-inspired learning rules have been used in the works from Troyer and Doupe (2000), Kröger et al. (2009) and Pagliarini et al. (2018a). Also, a Hebbian correlation rule involving the mean activation of neurons over a certain time interval (Sejnowski, 1977) has been used in the works from Westerman and Miranda (2002) and Oudeyer (2005). Otherwise, to define a learning rule one can use the distance between the target and the actual production in the goal space as in the work from Philippsen et al. (2016).

Reinforcement learning

Reinforcement learning (RL) is a mechanism to learn an action policy to maximize the expected reward, where the reward function encodes the goal. The goal space (internal representation of goals in Figure 2.1) defines the learning domain. The definition of the learning domain (given in Table 2.4 in column "*Perceptual space/Internal representation*") and of the reward function (given in Table 2.6 in column"*Evaluation*") are important to determine the complexity of the learning and the biological plausibility of the model.

Among the reviewed models there are models which implement classical RL: Troyer and Doupe (2000) used a plasticity rule which combines an associative learning rule and a reinforcement signal. Doya and Sejnowski (2000), Fiete et al. (2007) and Howard and Messum (2011) implemented reinforcement learning using a gradient ascent or descent algorithm. Similarly, gradient descent has been used in the work from Howard and Birkholz (2019). In these studies, the reward was computed as the correlation between spectrograms by Doya and Sejnowski (1998), or based on the feature of the song (the delayed estimation of the sum of the squares of pitch and amplitude) in the work from Fiete et al. (2007). In these models, the reward function, which is treated in Subsection 2.6.5, encodes the goal and contributes to the learning. Alternatively, the reward can be driven by the auditory salience as in the works from Warlaumont and Finnegan (2016), Howard and Messum (2007) and Philippsen et al. (2016), or by the caregiver choice which defines any novel situation that the agent must learn (Cohen and Billard, 2018).

Other authors did not choose to maximise a classical reward function but other quantities encoding the goal (e.g. a competence function, auditory salience). In this sense, RL has been implemented introducing intrinsically motivated exploration and active-goal selection. A Competence Progress algorithm which updates the internal representation of the goal and drives the exploration has been used by Moulin-Frier et al. (2014); Moulin-Frier and Oudeyer (2012): in the particular case of Moulin-Frier et al. (2014) the learning algorithm is based on Gaussian Mixture Models(GMM) updated via Bayesian inference in a self-supervised paradigm. Intrinsic motivation has been used by Forestier and Forestier et al. (2017) and different types of goal selection have been proposed by Moulin-Frier and Oudeyer (2012), Moulin-Frier et al. (2014). See Subsection 2.6.4 for details.

Optimisation algorithm

Learning can also be driven by an optimisation algorithm that tunes the motor parameters: this is an exception and hence we did not use a more general category *Parameter tuning* instead of *Learning* in Table 2.6. The optimisation procedure can aim to maximise

the ability of the agent in reproducing a selected goal via a reaching algorithm (Forestier and Oudeyer, 2017; Moulin-Frier and Oudeyer, 2012), or to maximise the reward (Cohen and Billard, 2018). Alternatively, a gradient inversion has been proposed by Bailly (1997), and a quasi-Newton gradient descent algorithm has been proposed by Howard and Messum (2011) to maximise a reward given by the combination of auditory salience, a diversity measure in the sensory space and en effort measure in the motor space. A particular example of optimisation algorithm is the Covariance Matrix Adaptation - Evolution Strategy (CMA-ES) (Murakami et al., 2015), that is a searching algorithm to maximise the confidence level of each goal. Finally, Najnin and Banerjee (2017) propose an actor-critic network to obtain the optimal sequence of actions to reach the target.

Supervised learning

Some works use supervised learning to learn the sensorimotor map. This could be implemented using an online algorithm via backpropagation as proposed by Liu and Xu (2014). Otherwise, this could be implemented combining backpropagation and gradient descent as proposed by Howard and Huckvale (2005). Supervised and unsupervised learning can also be used in combination with forward and inverse models, as in the work from Philippsen et al. (2014). They move from supervised self-training (thanks to the availability of a forward model) to unsupervised learning when imitation of novel contexts is included (after the training).

Other types of learning

Alternatively, incremental learning Gaussian Mixture Models (ilGMM) has been proposed by Acevedo-Valle et al. (2018) or GMM updated using Bayesian inference has been proposed by Moulin-Frier et al. (2014). A probability-based model has been proposed by Barnaud et al. with COSMO (Communicating Objects through SensoriMotor Operations) (Barnaud et al., 2019) architecture. This architecture was proposed by Moulin-Frier et al. (2015) and represents a Bayesian framework to approach vocal learning.

2.6.4 Exploration strategies

Different exploration strategies have been studied in the context of vocal learning or in other types of sensorimotor learning. Exploration can take place either in the motor space or in the goal space (perceptual space or internal representation). The simplest exploration mechanism is driven by uniform random exploration. Pure random exploration does not take into account (1) the memory of perceived stimuli (e.g. the distribution of percept vectors in the perceptual space), (2) the history of what has already been explored in the past. Several works use this approach to explore the motor space: Troyer and Doupe (2000), Westerman and Miranda (2002), Howard and Huckvale (2005), Howard and Messum (2007), Howard and Birkholz (2019), Oudeyer (2005), Moulin-Frier et al. (2015), Warlaumont and Finnegan (2016), Pagliarini et al. (2018a) and Barnaud et al. (2019). Alternatively, dynamic perturbation around a motor configuration has been used in the works from Doya and Sejnowski (1998) and Fiete et al. (2007) while implementing RL. A few authors used random exploration in the goal space: Forestier and Oudeyer (2017), Najnin and Banerjee (2017), Moulin-Frier and Oudeyer (2012) and Philippsen et al. (2014).

More sophisticated strategies are inspired by the nature of human development, which is progressive, incremental, autonomous and active. Behavioural analysis evidences how the actions of an agent are motivated by an internal or external reward. Following this idea, intrinsic motivation makes the agent choose an action basing the decision on the level of novelty, on the challenge it represents and on an internal reward. An example of such a strategy is called Intelligent Adaptive Curiosity (IAC) (Oudeyer et al., 2007): using a similarity-based progress maximisation the exploration is driven by the aim of maximising the learning progress, while the agent goes towards novel situations. Intrinsic motivation can drive motor babbling, defining a goal-directed exploration strategy. Usually, a competence function drives the choice of the next goal estimating the error or the reward or the level of knowledge relative to the goal. Different goal-directed strategies have been proposed in kinematic motor control learning by Forestier et al. (2017),

Forestier and Oudeyer (2016), Baranes and Oudeyer (2013) and Rolf et al. (2010).

Studies in the speech domain take inspiration from kinematic studies and introduce goal babbling as exploration strategy. This strategy allows the agent to do intermediate productions in the direction of the selected goal: that is, for any chosen goal the agent can define and make use of intermediate sub-goals to adapt the production. Goal babbling has been used by Liu and Xu (2014) and proposed in unsupervised learning driven by a measure of confidence to reproduce a sound as in the works from Philippsen et al. (2016) and Murakami et al. (2015), a competence progress as in the works from Moulin-Frier et al. (2014) and Moulin-Frier and Oudeyer (2012), an interest model as in the work from Acevedo-Valle et al. (2018) or the intrinsic reward as in the works from Forestier and Oudeyer (2017).

2.6.5 Evaluation

Evaluation of learning (or reward computation) can take place in the perceptual space, in the internal representation or in an additional space defined *ad hoc*. In models using the reinforcement learning (RL) paradigm it is common to have such *ad hoc* definitions: in such a case the evaluation is called *reward* and is computed by a *critic*. For example, the reward can be given by the correlation between the target and the output songs represented as a filtered, vectorized version of the sound spectrogram as in the work from Doya and Sejnowski (2000), or by the sum of the squares of pitch and amplitude as in the work from Fiete et al. (2007). In the work of Troyer and Doupe (2000), the quality of learning can be computed using the correlation coefficients between matrices representing the co-fluctuation of activity at different syllable epochs

In the case of intrinsically motivated agents, evaluation guides exploration, even if it does not contribute directly to the learning algorithm. These examples are related to evaluation computed in the goal space (i.e. perceptual or internal space). It can be computed using *competence progress* as proposed by Moulin-Frier and Oudeyer (2012) and Philippsen et al. (2016), or defining the confidence level of each goal as proposed

by Murakami et al. Murakami et al. (2015). Alternatively, other distance measures can be used to evaluate the learning in the perceptual space. For instance, Mean Square Error (MSE) has been used by Philippsen et al. (2014), an intensity measure has been used by Moulin-Frier et al. (2014), the distance between sound trajectories in the formant space is used by Forestier and Oudeyer (2017). Sum of Squared Error (SSE) has been used by Liu and Xu (2014) and Euclidean distance has been used by Acevedo-Valle et al. (2018) and Pagliarini et al. (2018a). A particular example of evaluation performed in the perceptual space is the work from Lyon et al. (2012) where a measure (called *F-measure*) is used to check the performance in learning the phonemes' dictionary.

Although evaluation is not usually implemented in the motor space, it is possible that some motor properties are used (e.g. articulator speed to compute the cost of a movement) to compute the reward. Kröger et al. (2009) compute the error value estimating the distance between the initial motor pattern and the estimated one. Interestingly, the works from Howard and Messum (2007, 2011) and Howard and Birkholz (2019) contain an example of a reward computed combining motor properties (voicing, effort, diversity) and auditory salience (computed using the spectral properties of the sound such as acoustic power, high to low frequency ration and vice-versa). Auditory salience has been used also in the work from Warlaumont and Finnegan (2016).

Two particular cases are given by the work from Howard and Huckvale (2005), where a spectrographic analysis is used to determine similarity between target and produced sound, and the work from Moulin-Frier et al. (2015) and Barnaud et al. (2019), where simulations are evaluated using the Dispersion Theory formula (Liljencrants et al., 1972) in the COSMO architecture (Moulin-Frier et al., 2015).

2.7 Discussion

To provide an accurate representation of the vocal learning process in humans or songbirds, a model should implement the biological mechanisms revealed by past experimental investigations at the behavioural, anatomical and physiological level (see the biological

context introduced in Section 2.2). The various models presented here are about song learning in songbirds and speech development in humans, but have been built to answer different questions (as highlighted in Section 2.3). However, we believe that comparing the various frameworks used to model different aspects of vocal learning will help to identify the important components and the links between them. Ultimately, such comparison may also reveal the next steps required to build a common model schema to study various questions about vocal learning and to account for a large number of experimental findings.

In Section 4.1 we introduced two kinds of sensorimotor learning models (see Figure 2.1 and Figure 2.2), the different spaces characterising a vocal learning model, and the functions going from one space to another. Table 2.4 and Table 2.6 highlight all the components we discussed in the review for all the considered models. However, it is not always possible to clearly identify each model component as they are missing in some models (indicated by "–" in the tables).

In general, motor control in vocal learning models is often based on pre-existing biologically-inspired models of vocal production and include the production of sound. The motor parameters are usually related either to sound properties (e.g. fundamental frequency, pitch period) or to anatomical parts of the body (e.g tongue and lips in humans, air pressure in birds). Models of sound production (e.g. VTL, DIVA) may not be able to reproduce perfectly the distribution of sounds that could be obtained from real data. Therefore, they may not have the same *perceptuo-motor phase space* than the target (e.g. infant's brain) they are trying to model. Indeed, the *perceptuo-motor phase space* is shaped by the fact that *"some regions of the motor command do almost not change the sound, while others change it abruply"* (**?**). Thus, we suggest that, in their computational experiments, modellers control for the potential discrepancy between produced sounds and target sounds. More generally, they should check for *perceptuo-motor phase space* discrepancies. Such an issue could impact learning efficiency.

Some learning frameworks do not take inspiration from biology. Indeed, some reinforcement learning algorithms and Hebbian learning rules used to implement synaptic plasticity are coherent with biology (as described in Section 2.2.5), but some authors

proposed biologically implausible learning algorithms (e.g. optimisation algorithms to implement trial-and-error strategies, or particular ways of training internal models). Moreover, it is not easy to cast learning algorithms into clear-cut categories: the ambiguity comes from the fact that different readers might have different definitions or categorisations. For instance, one can think about an architecture where a supervised algorithm is incorporated into a reinforcement learning framework.

The dimensions of the sensory, perceptual and motor spaces greatly vary among models, and the learning architectures do not deal with the same task complexity. Performance can thus not be directly compared between models. The choice of learning framework may constrain the authors to reduce the space dimensions: many learning frameworks and exploration strategies cannot deal with high-dimensional spaces, and brains likely reduce complexity because they cannot control all muscle fibers (Wolpert et al., 2001; Dhawale et al., 2017).

In order to find an evaluation strategy and reward function definition, it is convenient to have a low-dimensional preprocessed representation of the sound. To obtain such a representation, several reduction techniques have been used in the reviewed models: PCA and LDA (e.g. (Philippsen et al., 2016)), formant extraction (e.g. (Oudeyer, 2005; Moulin-Frier et al., 2014)), or scaling and normalization techniques (e.g. (Doya and Sejnowski, 1998; Kröger et al., 2009)). Ongoing studies try to use Variational Autoencoder (VAE) to help exploration strategies, reducing the goal space to a low-dimensional space while keeping an important part of the information encoded. For instance, Laversanne-Finot et al. (2018) use a particular type of VAE, called β-VAE to achieve this aim.

Models for sensorimotor learning with different motivations (e.g. grasping, recognition) are important complementary studies to take into account while studying vocal learning model. Indeed, these studies contain many important discussions about exploration strategies, target definition, motor space identification (Baranes and Oudeyer, 2013; Rolf et al., 2010) that can be useful to take inspiration for future investigation of vocal learning mechanisms. Perceptuo-motor skills, typical of speech production, do also exist in non-vocal gestures (Fowler, 2016). In some of the mentioned studies, other modalities

than vocal were used. For example, Forestier and Oudeyer (2017) propose two sensorimotor models: a vocal learning model and an action motor learning model. Cohen and Billard (2018) propose a model of symbol acquisition via active language learning (which combines vocal learning and symbol recognition).

We did not discuss previous modelling of the developmental aspects of vocal learning that investigate the effects of slow changes in the motor control apparatus or sensory system related to growth in the present review. It is, however, important to consider how such slow changes influence vocal production and interact with the learning process (Ghazanfar and Liao, 2018).

We provided different diagrams, tables, along with segmentation of spaces and functions, as a conceptual tool to analyse and compare existing models of vocal learning. We believe it provides several benefits: to understand the choices of the authors, to look at the biological plausibility of a model or part of it, to compare models systematically, and to give a baseline to build new models. We hope that researchers in the field will agree with our attempt of categorisations and comparisons, and that it will help in further studies to make descriptions more explicit and comparable.

Acknowledgment

This work was supported by the Inria CORDI-S PhD fellowship grant. We really want to thank the anonymous reviewers for all the pertinent remarks, questions and corrections they provided, which was of great help. We thank Jean-Luc Schwartz and Clément Moulin-Frier for helpful discussions. We thank Anthony Strock and Bhargav Teja Nallapu for proofreading the paper.

Chapter 3

A Bio-inspired Model Towards Vocal Gesture Learning in Songbird

The simplest vocal learning model includes only two spaces: the perceptual space (representing the auditory area of the bird's brain) and the motor space (representing the motor areas of the bird's brain). Moreover, it does not define sound production. As highlighted in Chapter 2, in this type of model the sensory and the motor space collapse in one space. To avoid any misunderstanding, in this chapter, the *sensory area* corresponds to the brain area where the sensory stimulus is encoded and the *motor area* corresponds to the brain area from where the input to the motor apparatus start. At the same time, as introduced in Chapter 2, the stimulus belongs to the sensory space, is encoded in the perceptual space, and is produced thanks to a motor command belonging to the motor space.

The model is inspired by the theoretical vocal learning model previously proposed by Hanuschkin et al. (2013) and Hahnloser and Ganguli (2013). The learning architecture is based on an inverse map from the sensory area onto the motor area, learned by a Hebbian learning rule. The exploration is random, and the sensory response enables the activations of sensory neurons. Here, the sensory response is referred to as the *auditory response*.

The chapter shows how it is possible to integrate bio-inspired assumptions in a theoretical inverse model for learning the connections between two neural populations. It

reproduces numerically the theoretical results and introduces the bio-inspired theoretical model of vocal learning in songbirds. A novelty non-linear sensory response function is introduced, and the Hebbian learning rule has been modified as a consequence. Moreover, it shows how the parameters can influence the learning progress: velocity and accuracy of learning are influenced by the selectivity on the one hand, and by the motor area size on the other hand. The latter will be the key point for further development of the model, as highlighted in Chapter 4. Section 3.1 introduces imitative sensorimotor learning and the model used as a starting point for the following. Section 3.2 describes the architecture of the model, highlighting the definition of the auditory response and learning rule. Section 3.3 shows the numerical implementation of the linear and non-linear models. In particular, it shows how the parameters can influence the velocity and the accuracy of learning. Section 3.4 summarizes the obtained results and gives some perspectives about how to develop this study.

"*A Bio-inspired Model Towards Vocal Gesture Learning in Songbird*" (Pagliarini et al., 2018a) has been published in ICDL-Epirob proceedings, and used for oral presentation and poster session multiple times. More details are available at https://github.com/spagliarini/2018-ICDL-EPIROB.

Abstract

The paper proposes a bio-inspired model for imitative sensorimotor learning, which aims at building a map between the sensory representations of gestures (sensory targets) and their underlying motor pattern through a random exploration of the motor space. An example of such a learning process occurs during vocal learning in humans or birds when young subjects babble and learn to copy previously heard adult vocalizations. Previous work has suggested that a simple Hebbian learning rule allows perfect imitation when sensory feedback is a purely linear function of the motor pattern underlying movement production. We aim at generalizing this model to the more realistic case where sensory responses are sparse and non-linear. To this end, we explore the performance of various

learning rules and normalizations and discuss their biological relevance. Importantly, the proposed model is robust whatever normalization is chosen. We show that both the imitation quality and the convergence time are highly dependent on the sensory selectivity and dimension of the motor representation.

Contents

3.1 Introduction

Imitative sensorimotor learning can be though as a control problem aiming to map the sensory input into a motor output. For example, humans and songbirds learn to produce

species-specific vocalizations as juveniles by imitating surrounding adults. The vocal learning process displays distinct (although partially overlapping) processes. First, during a sensory learning phase, young subjects memorize adult vocalizations, and build neuronal representations of their species vocal gestures. Then, in a sensorimotor phase, they start vocalizing and progressively converge to a good imitation of previously experienced vocalizations. It is believed that during the early phase of this sensorimotor process, called babbling, the subject maps representation of sensory (auditory) targets to the corresponding motor commands. In other words, the subject learns an inverse model. In this study, we assume the auditory selective responses to be already in place and investigate biological plausible mechanisms to learn an inverse model enabling this sensorimotor mapping.

An interesting property of some neurons in the brain is the ability to respond similarly during the production or observation (or listening for vocal gestures) of a given movement. This property has been linked with imitative sensorimotor learning and internal models (inverse or forward model). These neurons are referred to as mirror neuron (Oztop et al., 2006; Prather et al., 2008; Giret et al., 2014). While forward models describe a causal relationship between the sensory input and the motor system, inverse models have the aim to provide an appropriate motor command to a given state of the motor system. Although building an inverse model is easier when a forward model is available as proposed by Jordan and Rumelhart (1992) and Philippsen et al. (2014), inverse models could be enough to bootstrap the development of a simple computational mechanism, describing a memory system composed by the motor plan and the sensory stimulus (Oztop et al., 2013). The importance of computational models involving mirrors systems and learning processes has been stressed by Oztop et al. (2006, 2013), and the "Mirror-system hypothesis" stated by Arbib (2002) links mirror neurons with the emergence of human language during evolution. Songbirds learn their vocalization by imitation through a vocal learning process that very much resembles speech learning in human babies (Doupe and Kuhl, 1999). In the brain of songbirds, a part of the basal ganglia-thalamo-cortical circuitry is devoted to song learning in juveniles and plasticity in adults. This circuit is homologuous to the

basal ganglia circuits responsible for motor learning in mammals, and involved in speech learning in humans (Doupe et al., 2005; Mooney, 2009). Moreover, this song-related BG-thalamo-cortical circuit in birds receives input from mirror neurons (Prather et al., 2008). Therefore, the brain circuits responsible for avian song learning represent an ideal framework to study the neural mechanisms underlying imitative learning.

A theoretical model describing the implementation of an inverse model between auditive and motor areas through associative learning has been proposed recently (Hahnloser and Ganguli, 2013; Hanuschkin et al., 2013). The model is based on a simple Hebbian learning rule driven through random motor exploration and auditory feedback responses to this motor exploration. The proposed model assumes linearity for mathematical simplicity. As auditory responses in the brain are rather sparse and nonlinear we aim to extent the theoretical framework to a more realistic scenario (Hahnloser and Kotowicz, 2010).

We used an inverse model inspired by Hahnloser and Ganguli (2013) to describe the interaction between two populations of neurons, one formed by motor neurons and another formed by sensory neurons. We first show that replacing the learning rule by a simple normalized Hebbian learning rule allows rapid convergence in a simple non-linear model. We apply different normalizations in the learning rule. We then explore the influence of the sharpness of auditory selectivity in relation with the learning error after convergence and convergence time of the learning. Finally, we show how changing the number of sensory or motor dimensions modifies learning.

3.2 Method

3.2.1 Network and goal

The model includes two neural populations as shown in Fig. 3.1. The first layer is composed by afferent neurons and represents the sensory area. The second layer is composed by motor neurons, which represent the starting point for muscle activation, and thereby

for movement production. The synaptic weights describing the strength of the connection between neurons are defined by matrix W.

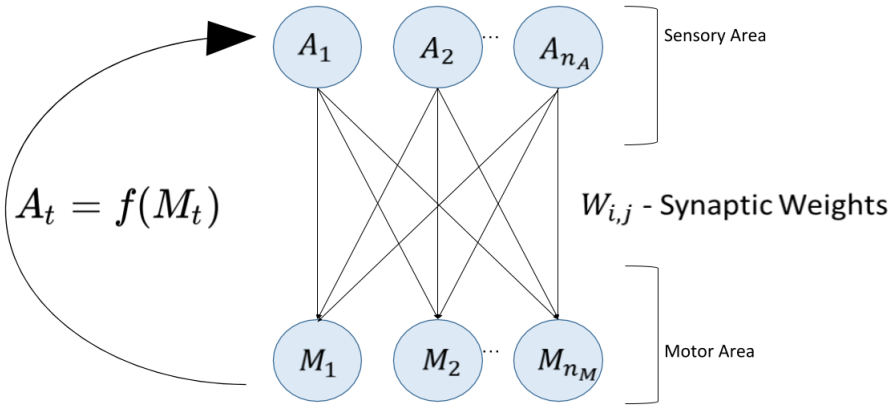


Figure 3.1: **Neural network schema.** The network includes two neural populations: the first layer is composed by n_a afferent neurons and represents the sensory area, the second layer is composed by n_m motor neurons and represents the motor area. $W_{i,j}$ represents the synaptic connections between sensory and motor neurons. Below the network schema we highlight how we define the sensory response. At each time step t, the sensory response is a function of the motor output, that is $A_t = f(M_t)$.

A model describing sensorimotor phase of learning based on a network as in Fig. 3.1 has been previously proposed by Hahnloser and Ganguli (2013). Neurons are linear units and at each time t motor and auditory activity are defined as a n_m-dimensional vector M_t and a n_a-dimensional vector A_t, where n_m and n_a represent respectively the number of motor and auditory neurons in the network.

3.2.2 Auditory response

At each time t the auditory activity A_t is defined as a function of the motor pattern M_t at that particular time, that is $A_t = f(M_t)$. Hahnloser and Ganguli (2013) define the auditory activity as a linear function of M_t, that is

$$A_t = QM_t, \tag{3.1}$$

where M_t represents the motor pattern at time t and Q the linear map defining the auditory activity due to the auditory feedback driven by current motor activity.

We then introduce non-linearity in the sensory response to auditory feedback. To represent selective responses as observed in various high sensory brain areas (e.g. auditory regions of the pallium in birds display responses selective to tutor syllables or to the bird's own syllables), we define the auditory activity A_t for each $j = 1, .., n_a$ neurons as a bell-shaped function around a target motor pattern:

$$A_{j_t} = exp\left(\frac{-||M_j^* - M_t||^2}{2\sigma^2 n_m}\right), \tag{3.2}$$

where σ represents the selectivity tuning width, n_m the number of motor neurons belonging to the network, M_t the motor pattern at time t and M^* the center of the auditory activity.

3.2.3 Learning process

Learning is driven by the Hebbian learning rule

$$\Delta W_t \propto \eta M_t A_t, \tag{3.3}$$

where W_t represents the synaptic weights between sensory and motor neurons, η the learning rate, M_t the motor pattern at time t and A_t the auditory activity at time t.

Synaptic weights $W_{t=t_0}$ between sensory and motor neurons are initially weak and increase according with (5.3) during learning until a certain time $t = t_f$, as

$$W_t = W_{t-1} + \Delta W_t, \tag{3.4}$$

where W_t represents the synaptic connections between sensory and motor neurons and ΔW_t is defined by (5.3).

However, synaptic weights have an upper boundary due to biological limitations (maximal number of synaptic receptors or neurotransmitters released). This can be introduced by a normalization either on the synaptic weights $W_{i,j}$ or on their variation $\Delta W_{i,j}$.

Hahnloser and Ganguli (2013) proposed a postdictive Hebbian learning rule given by

$$\Delta W_t = \eta(M_t - W_{t-1}A_t)A_t^T, \tag{3.5}$$

where W_t represents the synaptic weights between sensory and motor neurons, η the learning rate, M_t the motor pattern and A_t is defined by Eq. (3.1). Here the apex T indicates the transpose of the vector A_t.

In our model, we kept the basic Hebbian rule and we tested three normalizations in two different cases: a normalization over motor neurons (over all targets of one postsynaptic neuron) and a normalization over auditory neurons (over all inputs of one presynaptic neuron).

In practice, the two types of normalization are respectively implemented by normalizing over the lines or columns of the weights matrix W. The aim of the normalization is to bound either the mean of each column or line of $W_{i,j}$ or the euclidean norm of each column or line of $W_{i,j}$ to a maximum of 1. Considering the case of normalizing with respect to auditory neurons and pushing the mean of every column of W to a maximum of 1, the applied normalizations are the following:

- Maximum weights normalization

$$W_{i,j} = \frac{n_m W_{i,j}}{\Sigma_i W_{i,j}}, \tag{3.6}$$

- Supremum weights normalization

$$W_{i,j} = \begin{cases} W_{i,j} & \frac{\Sigma_i W_{i,j}}{n_m} < 1, \\ \frac{n_m W_{i,j}}{\Sigma_i W_{i,j}} & otherwise, \end{cases} \tag{3.7}$$

- Decreasing factor normalization

$$\Delta W_{i,j} = \eta M_t A_t \left(1 - \frac{\Sigma_i W_{i,j}}{n_m}\right). \tag{3.8}$$

Here $W_{i,j}$ represents the synaptic connections between sensory neuron j and motor neurons $i = 1, .., n_m$, where n_m is the number of motor neurons.

To obtain the normalization over the motor neurons and pushing the mean of each line of W to a maximum of 1, it is enough to change the index i for the index j and use the auditory dimension n_a. In order to use the norm instead of the mean in the definiton of the normalizations it is enough to introduce the norm on the column or the norm on the line of W in place of the mean.

At the same time, normalizing synaptic weights forces us to also normalize the motor target M^*, which represents what the model would learn at $t = t_f$. This is equivalent to a reduction of the target motor space (as it introduces a constraint on the final output of the model).

3.2.4 Simulation details

Each sensory neuron contains the response to a motor performance, and this is represented by the motor target M^*. We did not consider any strategy for the exploration. That is, at each time step t we simply considered the case of a random motor exploration M_t on which the auditory selectivity depends.

In the Hahnloser-Ganguli model there exist a direct map which defines the motor activity from the auditory activity as $M_t = Q^d A_t$. For the inverse model we need to define A_t in dependence on M_t as in Eq. (3.1). That is the inverse matrix of Q, i.e Q^d represents the motor target M^*, that is the ideal motor activity which the model should have learned once learning phase has ended. The goal then is to activate each sensory neuron can drive M_j^* through W, such that

$$WA^* \longrightarrow M^*, \tag{3.9}$$

where $A^* = A_0 I$ defines the ideal auditory activity. We fixed $A_0 = 1$.

Given the motor targets M^*, at each time step t, the distance between what the model actually learned and what it should have learned is defined as

$$d_t = \frac{||M^* - W_t A^*||}{n_m}, \tag{3.10}$$

where M^* represents the motor target, W_t the synaptic weights matrix, A^* the ideal auditory activity and n_m the number of motor neurons.

We defined the convergence time τ as the number of time steps at which the updates of the weights are small enough. That is, the distance between what the model targets should have learned and what he effectively learned reaches a plateau. After have chosen

$\epsilon = 1$ at $t = t_0$, given an interval of time $\Delta t = [k, k + 2N]$ of measure $2N$, we defined

$$\epsilon = \frac{1}{2N}\left(\Sigma_{k+N}^{k+2N} d_t - \Sigma_k^{k+N} d_t\right), \tag{3.11}$$

and we used it as threshold, in a way that a particular experiment stops either if ϵ reaches the value $\epsilon^* = 10^{-9}$ either if it goes until a fixed time $t = t_f$. We tested several values for the tuning selectivity width, i.e. $\sigma = [0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7]$, varying in this way the auditory selectivity. We used for almost every value an interval of length $N = 400$. However, since the distance evolves very slowly, for small values of σ this interval is not large enough. For instance, an interval with $N = 500000$ has been used for $\sigma = 0.02$.

3.3 Results

3.3.1 Simple model with linear auditory responses

Fig. 3.2 shows the evolution in time of the smooth average distance between WA^* and the target motor pattern M^*, defined by Eq. (3.10) for each simulation. In the linear version of the model (blue line) , the auditory activity is a linear function of the motor production, as in Eq. (3.1), and the postdictive Hebbian rule defined by Eq. (3.5) guides learning. As expected by the theory (Hahnloser and Ganguli, 2013), the distance between WA^* and the target motor pattern M^* (which is the inverse of Q) converges exponentially to zero. The learning rule therefore ensures proper learning of the inverse model. In contrast, when auditory feedback is non-linear (orange line in Fig. 3.2), where the auditory activity is defined by Eq. (3.2), the postdictive Hebbian learning rule does not allow convergence, and the distance between WA^* and the target motor pattern M^* rather diverges.

Define the auditory activity as in Eq. (3.1) means that the matrix Q needs to be invertible to reach convergence. We also don't know how to define the exploration space starting from the map Q. We assumed as given the space of the inverse of Q, which means that we solved at each time step t the linear system $Q^{-1}A_t = M_t$, where $Q^{-1} = M^*$. At the same time to solve a linear system means to face the problem of invertibility of matrix M^*. Indeed, if the matrix M^* is singular or close to be singular, then the numerical

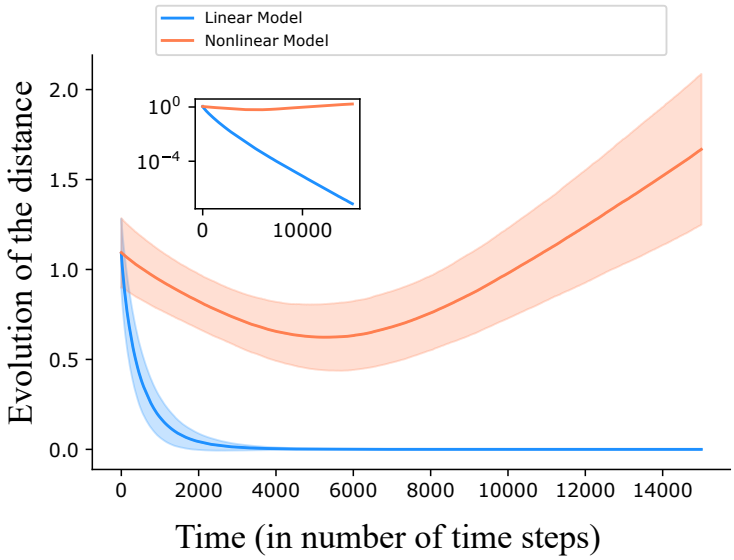


Figure 3.2: **Ganguli-Hahnloser linear and nonlinear model.** Evolution in time of the smooth average distance between WA^* and the target motor pattern M^*, computed over 50 simulations. Comparison between the Hahnloser-Ganguli linear model (in blue) where the auditory response is defined by Eq. (3.1) and the nonlinear version of Hahnloser-Ganguli model (in orange) where the auditory response is defined by Eq. (3.2). In both cases learning is driven by the postidictive Hebbian learning rule in Eq. (3.5) and weights are updated following Eq. (3.4). To highlight the behavior of the linear model, the same comparison using a log scale is shown in the box. Parameters value: $t_f = 1.5 * 10^4$, $n_m = n_a = 3$, $\eta = 0.01$, $\sigma = 0.1$, $C = 20$.

algorithm is not longer working. To avoid an ill-posed problem we added a condition by computing the condition number[1] of the matrix M^*, forcing it to be such that

$$k(M^*) < C, \tag{3.12}$$

where M^* represents the motor target and C is a positive constant belonging to $[1, +\infty]$. In this way we simulated all the simulations in Fig. 3.2 without having ill-posed problems. Without the application of this condition, simulations were often diverging because of the divergence of $M^{*^{-1}}$. A nonlinear auditory response defined by Eq. (3.2) enables to avoid

[1]Given a general linear system $Ax = b$, its condition number is defined as $k(A) = \xi_{max}(A)/\xi_{min}(A)$, where $\xi_{max}(A$ and $\xi_{min}(A)$ are respectively the maximal and minimal singular values of A. The value $k(A)$ represents the variability of the solution, so how much the solution x changes consequently to a change in b. The lower bound for the condition number is $k(A) = 1$, whereas it can reaches the value $k(A) = \infty$ if the matrix A is singular.

ill-posed problems. At the same time (as underlined in the small box with logarithmic scale in Fig. 3.2) it leads to a divergence in the distance between the target motor pattern M^* and WA^*. That is, the model does not learn anymore after the introduction of nonlinearity. So far, instead of keeping the postdictive Hebbian rule proposed by Hahnloser and Ganguli (2013), we used a traditional Hebbian rule to drive learning and tried to face the problem by applying other types of normalization.

3.3.2 A nonlinear auditory response

Fig. 3.3 shows the evolution of the distance between WA^* and the target motor pattern M^* for one example neuron. The initially weak synaptic connections evolves following the Hebbian learning rule given by Eq. (5.3) and finally approaches M^*. To obtain these results we applied the normalization defined by Eq. (3.8). By introducing nonlinearity, as shown by Fig. 3.2, the Ganguli-Hahnloser model does not converge anymore.

We tested three different normalizations, given by Eq. (3.6), Eq. (3.7) and Eq. (3.8). The upper panel of Fig. 3.4 shows the comparison between the three normalizations with respect to auditory neurons. That is, with respect to the columns of W. Normalizations given by Eq. (3.6) and Eq. (3.7) are applied directly to the weights matrix, which gives a faster convergence but a lost in smoothness. Normalization given by Eq. (3.8) is applied to the variation of the weights, by multiplying its classical definition by a decreasing factor. This means that the variation is smaller and smaller as the weights approaches the target, which results in a smooth trend of the distance curve. To highlight better the difference between normalization over auditory and motor neurons, the bottom panel of Fig. 3.4 shows the comparison between the normalization given by Eq. (3.8) applied in its mean and norm version. As it is shown, a normalization over auditory neurons works better than the same normalization over motor neurons in the sense that the distance between WA^* and M^* is lower if the normalization is applied over the auditory neurons than if the normalization is applied over the motor neurons.

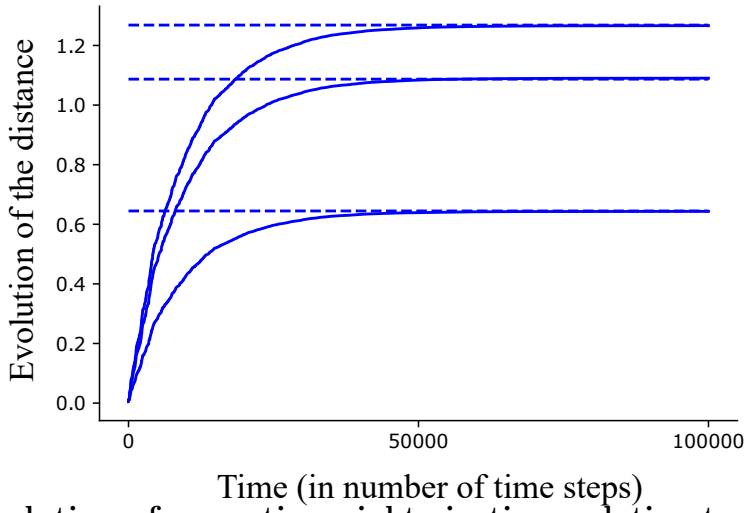


Figure 3.3: **Evolution of synaptic weights in time relative to a single auditory neuron and distance from the target motor pattern (three neurons example).** Evolution in time of the synaptic weights W (continuous blue line) and the target motor pattern M^* (dashed blue line) for one example neuron. Each auditory neuron is composed by three components represented by the three lines. Here the auditory activity is defined by Eq. (3.2) and learning is driven by the Hebbian learning rule in Eq. (5.3). At each time steps weights are updated following Eq. (3.4) and the normalization defined by Eq. (3.8) has been applied. Parameters value: $t_f = 1 * 10^5$, $n_m = 3$, $n_a = 1$, $\eta = 0.01$, $\sigma = 0.1$.

3.3.3 Auditory selectivity effect

Auditory selectivity impact on the learning can be observed by varying its value and by observing both the convergence time τ both the distance at $t = \tau$ between WA^* and M^*. Fig. 3.5 shows how the mean convergence time τ and the mean distance d_t at $t = \tau$ depends on the auditory selectivity. As the tuning selectivity width σ increases, a decreasing in the mean convergence time and an increasing in the mean distance can be observed. In particular, for the value $\sigma = 0.02$ convergence time is not fully correct because many simulation reached a fixed time $t_f = 2 * 10^7$ before having reached convergence. This is displayed on the plot by the first dashed part of the red line.

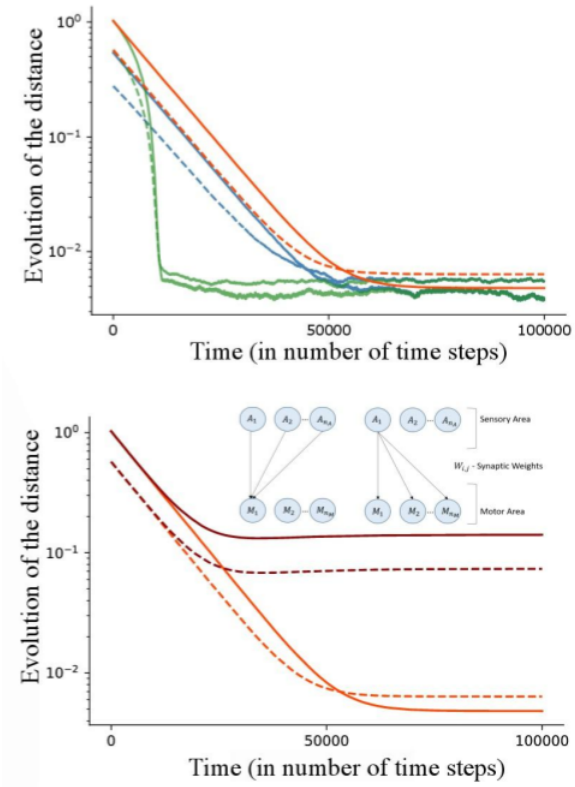


Figure 3.4: **Comparison between different types of normalization: evolution in time of the distance.** Evolution in time of the smooth average distance between WA^* and the target motor pattern M^*, computed over 50 simulations. (Top) Comparison between the model normalized by the maximum weights normalization in Eq. (3.6) and the corresponding norm version (respectively the continuous blue line and the dashed blue line), the model normalized by the supremum weights normalization in Eq. (3.7) and the corresponding norm version (respectively the continuous green line and the dashed green line), the model normalized by the decreasing factor normalization in Eq. (3.8) and the corresponding norm version (respectively the continuous red line and the dashed red line). All the normalizations have been taken with respect to auditory neurons. (Bottom) Comparison between the model normalized by the decreasing factor normalization in Eq. (3.8) with respect to auditory neurons (red lines) and with respect to motor neurons (dark red lines). Comparison between the normalization applied using the mean of W (continuous lines) and the norm of W (dashed lines). Here the auditory activity is defined by Eq. (3.2) and learning is driven by the Hebbian learning rule in Eq. (5.3). At each time steps weights are updated following Eq. (3.4). Parameters value: $t_f = 1 * 10^5$, $n_m = n_a = 3$, $\eta = 0.01$, $\sigma = 0.1$.

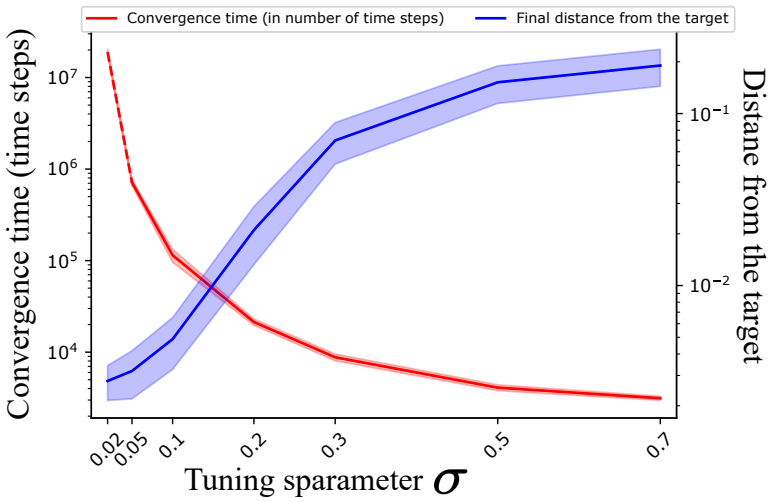


Figure 3.5: **Effect of the auditory selectivity on convergence time and distance.** Auditory selectivity impact on the convergence time (in red) and on the distance between $WA*$ at the convergence time τ and the target motor pattern M^* (in blue), computed over 50 simulations. The first dashed part of the red line underlines the fact that for $\sigma = 0.02$ not all the simulations converges before having reach a fixed simulation exit time $t_f = 2 * 10^7$. Here the auditory activity is defined by Eq. (3.2) and learning is driven by the Hebbian learning rule in Eq. (5.3). At each time steps weights are updated following Eq. (3.4). Parameters value: $\sigma = [0.02, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7]$, $n_m = n_a = 3$, $\eta = 0.01$, $\epsilon^* = 10^{-9}$. We applied the decreasing factor normalization given by Eq. (3.8). To exit the simulations we compute ϵ as in Eq. (3.11). We used for almost every value an interval with $N = 400$. However, since the distance evolves very slowly, for small values of σ this interval is not large enough. For instance, an interval with $N = 500000$ has been used for $\sigma = 0.02$.

3.3.4 Varying network dimensions

The quality of learning in terms of the distance at the convergence time $t = \tau$ and how slow learning develops can be investigated by varying the network dimension. Firstly, the number of auditory neurons has been kept fixed at the value $n_a = 3$, and the number of motor neurons varied as $n_m = [2, 3, 4, 5, 6, 7]$. Then, viceversa, the number of motor neurons has been kept fixed at $n_m = 3$ and the number of the auditory neurons varied using the same values as before. Fig. 3.6 shows the effect of changes in the network dimension respectively on the mean convergence time τ and on the mean distance at $t = \tau$, computed over 50 simulations. The upper panel shows how, keeping fixed the motor

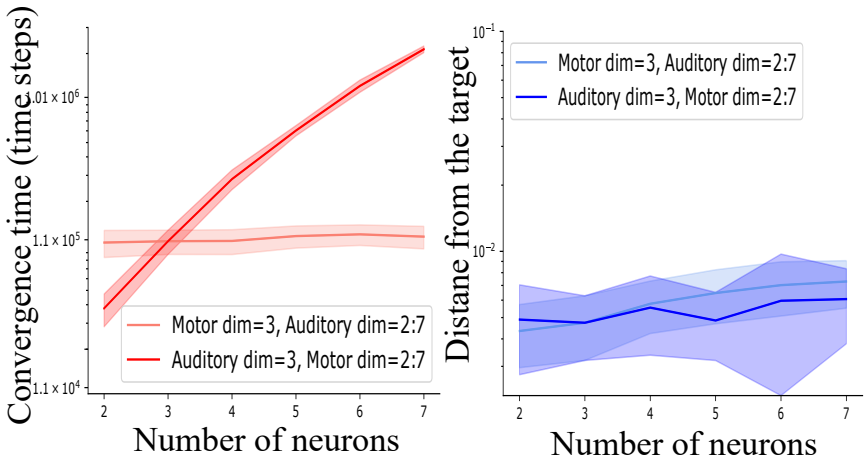


Figure 3.6: **Effect of network dimension on the convergence time and the distance.** (Left) Network dimensions effect on the mean convergence time τ and (Right) on the mean distance between WA^* at time $t = \tau$ and the target motor pattern M^*, computed over 50 simulations. The dark red line refers to a network where the number of auditory neurons has been kept fixed at $n_a = 3$, whereas the light red line refers to a network where the number of motor neurons has been kept fixed at $n_m = 3$. In both cases, the second dimension has been varied as $[2, 3, 4, 5, 6, 7]$. Parameters value: $\eta = 0.01$, $\sigma = 0.1$. Here we applied the decreasing factor normalization given by Eq. (3.8). To exit the simulations we compute ϵ as in Eq. (3.11) with $N = 400$.

dimension, there is not an evidence of network dimension effect on the mean convergence time. Viceversa, keeping fixed the auditory dimension the learning slows down as the motor dimension increases. However, the mean distance at $t = \tau$ is not affected by any change in the neural network dimensions, as shown in the bottom panel of Fig. 3.6. More details are available at https://github.com/spagliarini/2018-ICDL-EPIROB.

3.4 Discussion

Hahnloser and Ganguli (2013) proposed a simple mathematical framework to approach the sensorimotor learning problem in songbirds. It is based on a linear auditory activity and a postdictive Hebbian learning rule. Linearity in the auditory activity makes the theoretical investigation possible but is not biologically realistic. To be invertible, the matrix Q for

auditory response must be squared, which means that auditory and motor dimensions have to be equal. Moreover, numerical implementation of the learning algorithm proposed by Hahnloser and Ganguli requires to invert the auditory response matrix (Q) to determine the range of motor output required for proper exploration and learning. As there is no general method to invert a random matrix Q, a numerical implementation of the model requires to set a specific Q that can be inverted. Finally, the postdictive learning rule only works for the linear model and it is not clear whether biologically realistic learning rules can still lead to convergence or near-convergence in the case of non-linear auditory feedback.

As Hebbian or associative learning rules are implemented through activity-dependent synapse-specific increases in synaptic weights (synaptic potentiation), that must be augmented by global processes that regulate overall levels of neuronal and network activity to prevent explosion of synaptic weights (Abbott and Nelson, 2000). Regulatory processes are often as important as the more intensively studied Hebbian processes in determining the consequences of synaptic plasticity for network function. Setting an upper bound on the total synaptic weights to or from a given neuron may also reflect the biological limitation of synaptic connections: limits are imposed on their growth due to the limited quantity of available material (receptors, neurotransmitters, ...). The introduction of normalization on the weights or on their variations is a simple solution to this problem. Several forms of normalization were considered here to take into account this biological limitation. While the linear model of Hahnloser and Ganguli (2013) converges for the postdictive learning rule described there, we show that near-convergence can be achieved with multiple normalization rules added to a simple and typical associative learning rule given by Eq. (5.3). Convergence time, final distance from motor output to target, and smoothness of the distance evolution through time all depend on the specific normalization used. However, it is important to notice that the final "error" (distance from motor final weights to target motor pattern) does not vary much with normalization, assuming it is applied on all synaptic outputs from a given presynaptic (auditory) neuron.

In most of the simulations we focused on normalization given by the decreasing factor

normalization in Eq. (3.8). We kept this normalization for all our analyses because it gives better performance (i.e. low "error"). We noticed that when this normalization is applied, it gives better performances over auditory neurons (presynaptic) than over motor neurons (postsynaptic), despite the fact that a normalization with respect to motor neurons (regulated at the level of the post-synaptic neuron) may be more biologically plausible. Although other forms of plasticity exist, including presynaptic modulation of synaptic strength, classic long-term potentiation/depression (LTP/LTD) mechanisms mostly involve postsynaptic receptor reorganization.

A remaining open question is related to the final value (after convergence) of the distance between the target motor pattern and what the model actually learned. Future work is needed to determine the factors that determine this final error and how it can be reduced. One possibility is that various motor targets (one for each auditory neuron) may interfere during learning, leading to imperfect copies. However, our preliminary experiments didn't show that interference had an influence on the final error.

We investigated how learning depends on the auditory selectivity observing that as the tuning selectivity width σ increases as the final distance between weight matrix and motor target (error) increases while convergence time decreases. There is therefore a trade-off between learning speed and accuracy that can be balanced through the selectivity of auditory neurons. One way to make learning both fast and accurate could be to start the learning process with a large tuning width (low selectivity) and to decrease it progressively as learning goes on. Interestingly, in many songbird species (including the well-studied zebra finches), sensory learning overlaps with sensorimotor learning, and the auditory selectivity therefore develops during the early sensorimotor phase.

Finally, taking into account the influence of the dimensions (i.e. number of units) of the sensory and motor layers we noticed that motor dimension has a strong influence on convergence time. This strong influence comes from the fact that for lower values of σ the distance between the target and WA^* tents to decrease much more slowly. In our network we are considering a motor output that does not distinguishes muscle control and sound production. It is not clear which of these two components is responsible for the high

increase in convergence time. A model displaying a motor output and a sound generating system is needed to resolve this question. Future work could include (1) the addition of an artificial syrinx model as motor output and more auditive like feature selectivity in the auditory layer, and (2) the influence of different exploration methods such as goal-directed exploration.

Acknowledgment

We would like to thank Camille Soetaert and Jean-Baptiste Zacchello for preliminary work done. We also thank Inria for the CORDI-S PhD fellowship grant.

3.5 Non-published complementary results

3.5.1 Babbling generation

In this chapter, we built the model assuming that at each time step a new motor exploration takes place, a new auditory response is computed and the synaptic weights are updated as a consequence. That is, syllables has been considered as entities lasting the time of one-time step, without taking into consideration the fact that biologically they do have a certain duration. Moreover, the delay between the motor neurons activity and the activity of the auditory neurons (which causes the overlap between the auditory representation of a syllable and the production of the new syllable) has not been taken account.

To introduce a babbling paradigm in the model means to introduce the concept of the syllable (continuous lines in the top panel of Figure 3.7) and gap (dotted lines in the top panel of Figure 3.7) duration. Mathematically, this means that for a certain number of timesteps the same motor exploration m_i is performed and the same auditory response a_i is received. To obtain such a babbling paradigm, we introduced the syllable duration, the gap duration, and the initial delay as values taken from an exponential distribution

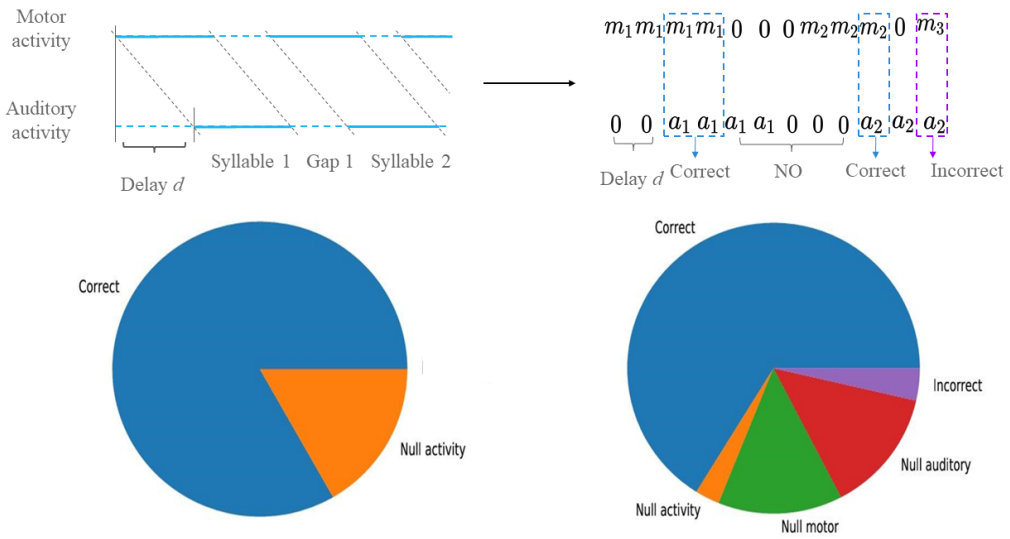


Figure 3.7: **Babbling paradigm**. The top panel shows the schema of the motor and the auditory activity (left) and how it translates in term of each motor exploration m_i and its corresponding auditory response a_i (right). The continuous lines in the activity represent the syllables, whereas the dot lines represent the gap between two consecutive syllables. The parameter d represents the delay between the motor neuron activity and the activity of the auditory neurons. The continuous lines in the activity represent the syllables, whereas the dot lines represent the gap between two consecutive syllables. The bottom panels represent the possible configurations that can arise from the activity: correct (correspondence between the motor activity and the auditory activity), incorrect (i.e., mismatched, when there is no correspondence between the motor activity and the auditory activity), null motor (motor activity equal to zero), null auditory (auditory activity equal to zero), null activity (both motor and auditory activity equal to zero). If $d = 0$, only correct and incorrect configurations are present. Otherwise, depending on the activity parameters λ_d, λ_{gap} and λ_{syll}.

of parameter$\lambda > 0$ (Darshan et al., 2017):

$$f(x) = \lambda e^{-\lambda x} \qquad wherex \geq 0 \tag{3.13}$$

The syllable duration follows an exponential law of mean $\lambda_{syll} = 150$ ms (i.e., 15 time steps). The gap duration follows an exponential law of mean $\lambda_{gap} = 50$ ms (i.e., 5 time steps). The delay duration follows an exponential law of mean $\lambda_d = 30$ ms (i.e., 3 time steps).

In terms of time steps, each motor exploration m_i and its corresponding auditory response a_i last for several time steps: for instance, in the top panel of Figure 3.7 the motor exploration m_1 lasts 4 time steps, whereas the motor exploration m_2 lasts 3 time steps. Taking into account the delay d and the gap between syllables, the four possible configurations that can emerge are listed below.

- Correct: when there is a correspondence between the motor activity and the auditory activity (i.e., m_i happens at the same time step as a_i).
- Null motor: when the motor activity is equal to zero, but the auditory activity is not equal to zero.
- Null auditory: when the auditory activity is equal to zero, but the motor activity is not equal to zero.
- Null activity: when both the motor and the auditory activities are equal to zero.
- Incorrect: when both the motor and the auditory activities are different from zero and there is not a correspondence between them (i.e., m_i happens at the same time step as a_j, with $i \neq j$).

Temporally, as shown in the bottom-left panel, m_i and a_i coincide only when $d = 0$ (i.e., there is no delay between the motor activity and the auditory activity). Indeed, when $d = 0$, the null activity corresponds exactly to the occurrence of gaps. The null activity here corresponds to the occurrence of the gap. Otherwise, when $d \neq 0$ the percentage of correct correspondences is lower, but it remains higher than the percentage of null or incorrect configurations. Else, if $d \neq 0$, several configurations can be observed: correct correspondence between motor and auditory activity (i.e., m_i happens at the same time step as a_i); incorrect (i.e., m_i happens at the same time step as a_j, with $i \neq j$); null activity (when the gap in the motor activity corresponds to a gap in the auditory activity); null motor or auditory (when either the motor or the auditory activity is equal to 0, but the other is different from 0). The bottom left diagram of Figure 3.7 shows an example

summary of how many times each of this configuration happen, over a simulation lasting up to a final time $t_f = 3 * 10^5$.

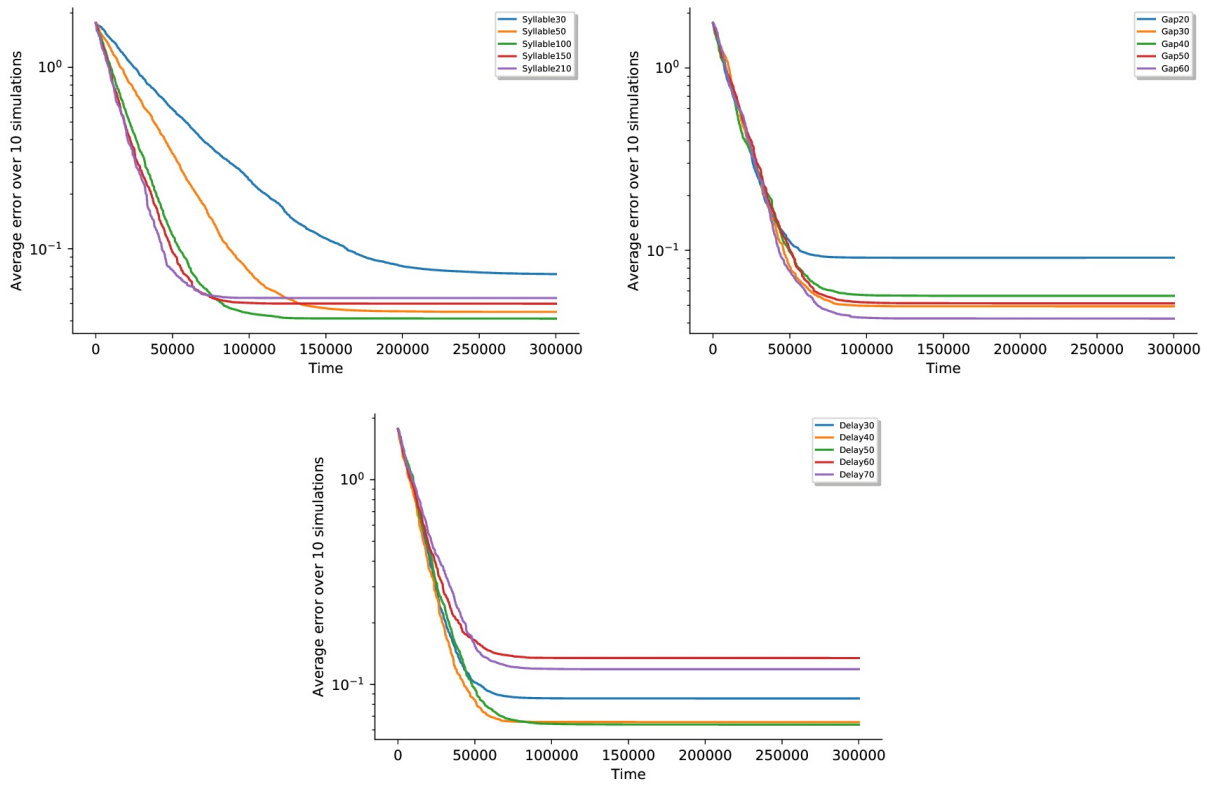


Figure 3.8: **Comparison between different parameters of the syllable, gap and delay distribution.** Evolution of the distance depending on the parameters λ_{syll} (top left panel), λ_{gap} (top right panel) and λ_d (bottom panel). To observe the dependence on the syllable duration, the parameters relative to the distribution of the gap and the delay have been kept fixed (i.e., $\lambda_{gap} = 50$ ms and $\lambda_d = 30$). The parameter λ_{syll} has been varied as $[30, 50, 100, 150, 210]$. To observe the dependence on the gap duration, the parameters relative to the distribution of the syllables and the delay have been kept fixed(i.e., $\lambda_{syll} = 150$ ms and $\lambda_d = 30$). The parameter λ_{gap} has been varied as $[20, 30, 40, 50, 60]$. To observe the dependence on the delay duration, the parameters relative to the distribution of the syllables and the gap have been kept fixed(i.e., $\lambda_{syll} = 150$ ms and $\lambda_{gap} = 50$). The parameter λ_d has been varied as $[30, 40, 50, 60, 70]$. Parameters value: $t_f = 3 * 10^5$, $n_m = n_a = 3$, $\eta = 0.01$, $\sigma = 0.1$

.

We kept the same network dimensions as in the model introduced in the main chapter, with $n_m = n_a = 3$. We used an Hebbian learning rulle normalizes using Equation 3.8 (the

one that was working better for the simple task). We fixed the learning rate at $\eta = 0.01$, the auditory selectivity at $\sigma = 0.1$ and the exit time at $t_f = 3 * 10^5$. The babbling paradigm introduces three new parameters: λ_{syll}, λ_{gap} and λ_d. For each of these three parameters, we explored different values keeping fix the others. The aim is to observe how the distance between the target and the ideal motor pattern evolves depending on each of the three new parameters.

3.5.2 Results

The evolution in time of the smooth average distance between WA^* and the target motor pattern M^* shows a higher plateau for the error when small values of λ_{syll} (top left panel of Figure 3.8) and λ_{gap} (top right panel of Figure 3.8) or high values of λ_d (bottom panel of Figure 3.8) are used.

At convergence (i.e., at time $t_f = 3 * 10^5$), the average final distance between WA^* and the target motor pattern M^* depends on the combination of the syllable and gap duration. As shown in Figure 3.8, small values for λ_{syll} and λ_{gap} lead to high distance values. A small λ_{syll} does not help the learning efficiecy (Figure 3.9): the bottom line of the matrix contains values greater than 0.08 independently from the gap duration. Increasing both the gap and the syllable duration parameters, things get better: if $\lambda_{syll} \geq 100$ ms and $\lambda_{gap} \geq 30$, the distance remains smaller than 0.04. Interestinlgy, the lower average distance has been observer for $\lambda_{syll} = 50$ and $\lambda_{gap} = 30$.

3.5.3 Discussion

The delay of the auditory feedback plays an important role in how a sequence of syllables is perceived, introducing uncertainty in the predictions of syllables (Bouchard and Brainard, 2016). Indeed, sensory feedback impacts the repetitions of motor sequences by enabling long repetitive vocal sequences (Wittenbach et al., 2015). Similarly, during speaking, speech-articulator representations are temporally coordinated in humans (Bouchard et al., 2013). Indeed, the motor command and the sensory information are combined to gather

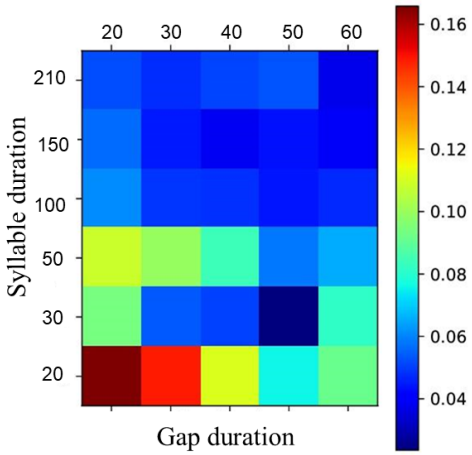


Figure 3.9: **Final average distance: comparison between different parameters for the syllables and the duration of the gaps.** Each square represent the average distance between WA^* and the target motor pattern M^* computed over 10 simulations, at time $t_f = 3 * 10^5$. Here, the axis represents the variation of the syllable duration parameter λ_{syll} (vertical axis), and the variation of the gap duration parameter λ_{gap} (horizontal axis). To observe how the final distance depends on the combination of syllable and gap duration, the parameters relative to the distribution of the delay have been kept fixed(i.e., $\lambda_d = 30$ ms). Instead, λ_{syll} and λ_{gap} have been varied, respectively, as $[20, 20, 50, 100, 150, 210]$ and $[20, 30, 40, 50, 60]$. The color scale becomes reddish as the error increases. Lower errors are achieved when λ_{syll} assumes low values, λ_{gap} assumes high values, λ_d assumes low values. Parameter values: $t_f = 3 * 10^5$, $n_m = n_a = 3$, $\eta = 0.01$, $\sigma = 0.1$

a rapid feedback control (Guenther et al., 2006b). A negative feedback can play a role in speech disorders such as stuttering (Wittenbach et al., 2015).

The model proposed in Section 3.5 describes the overlap between motor activity and auditory activity which has been shown experimentally (Bouchard and Brainard, 2013; Darshan et al., 2017). Such an overlap is caused by the presence of a delay in the auditory feedback, which usually has the same order of the duration of the syllables (CITE?). Indeed, if the delay distribution is kept fix ($\lambda_d = 30$ ms), it is possible to observe how long syllables do not allow a high learning accuracy (bottom line in Figure 3.9).

Chapter 4

What does the Canary Say? WaveGAN Applied to Birdsong

Abstract

Speech generation is a complex problem which has been approached by several studies. Generative Adversarial Networks (GANs) have shown very good abilites generating images, and more recently sounds. An example is given by WaveGAN. We aim to test the ability of WaveGAN to produce realistic canary syllables under the condition of having a small latent space dimension. We first trained WaveGAN varying the latent space dimension from 1 to 6 on a dataset of 16 different canary syllables. We show that a latent space of dimension 3 is enough to produce sounds of quality often indistinguishable from real canary ones, while reproducing all the types of syllables of the dataset. Then, we built a RNN-based classifier able to recognise the syllables of the dataset. Afterwords, we use this classifier to identify the generated samples. We measure both quantitatively and qualitatively the output across GAN training epochs and latent dimension. We also compare different instances of training. Importantly, we show that a 3-dimensional GAN is able to both reproduce the syllables and generalise by interpolating between the syllables. We used UMAP to qualitatively show the similarities between the training data and

the generated data, and between the generated syllables and the interpolations produced. Interestingly, this study provides tools to train simple sensorimotor models, as inverse models, from perceived sounds to motor representations of the same sounds. Both the RNN-based classifier and the small dimensional GAN provide a way to learn the mappings of perceived and produced. This chapter will be submitted for pubblication soon.

Contents

4.1 Introduction

The presence of different talkers, rates and contexts makes speech composition difficult to define physically. Indeed, when different talkers produce the same sound, the acoustics (i.e. formant frequencies) vary (Hillenbrand et al., 1995), making speech highly variable. Similarly, faster speech acoustics differ from slower speech acoustics, and contexts variations can determine a change in the acoustic features (Miller and Liberman, 1979). That is, speech represents a high dimensional domain difficult to deal with. As a consequence, speech generation is a challenging problem which involves several research fields such as signal processing and machine learning.

Speech generation and vocal learning face some common challenges due to the interpretability and the complexity of the data they are dealing with. Moreover, besides the subject of the study, a vocal learning model may require a realistic (i.e. a vocal output that belongs to the probability distribution of the real training dataset) vocal production apparatus. That is, a motor control function is necessary to obtain a complete vocal learning model (Pagliarini et al., 2020). Behavioural and the neuroanatomical similarities between songbirds and humans (Kuhl, 2004; Chakraborty and Jarvis, 2015) suggest songbirds as a model for a *reduced* version of the vocal learning in humans, enabling simpler

hypotheses. In particular, canaries have a large highly variant repertoire and their songs are characterized by a complex syntax with long-time dependencies (Markowitz et al., 2013). These properties make canary songs a reasonable middle ground between human speech and simpler birdsongs (e.g., zebra finches' song).

We aim to investigate if a generative model can be used as the motor control function in a vocal learning model describing sensorimotor learning. In such a scenario, the generative model plays the role of the motor control function and takes the motor space as input. At the same time, the motor space is involved in the learning algorithm: a high dimensional motor space would make the learning computationally costly (Pagliarini et al., 2018a). Some models based on dynamical systems are interesting candidates as motor generator models (Gardner et al., 2001; Amador et al., 2013).

Generative Adversarial Networks (GANs) are an example of generative models which enable to represent high-dimensional distributions in a latent space. In general, the main components of a GAN are a generator, that tries to reproduce a target distribution (e.g. images) given random inputs, and a discriminator, which tries to recognize real samples from generated ones. The "latent space" of a GAN is the input dimension of the generator part after training: continuous changes in this latent space will hopefully produce smooth changes in the generated samples. GANs have been used to produce complex datasets such as images, sounds, music, speech and, to a lesser extent, birdsongs (Salimans et al., 2016; Dong et al., 2018; Donahue et al., 2018). Indeed, Donahue et al. (2018) trained their model, called WaveGAN, on a speech dataset composed of 10 classes (i.e., 10 words representing the numbers from 0 to 9) and on wild recordings from several bird species. The promising but noisy results on a wide and highly variable dataset of birdsongs determined our choice to study the performance of the WaveGAN generator on a smaller, cleaner dataset, more similar to the speech dataset used in the original work.

In this paper, we trained a Wesserstein GAN with Gradient Penalty (WGAN-GP) (Arjovsky et al., 2017) using the WaveGAN (Donahue et al., 2018) setup on an adult canary dataset. The main objectives of this study is not to improve the current version of WaveGAN, but rather to test its ability to produce realistic canary syllables for different

conditions of two parameters: the latent space dimension and the size of the dataset. On the one hand, we are interested in finding the minimal latent space dimension that allows to reproduce real songs fluctuations. On the other hand, we are interested in exploring the capability of the generator to reproduce a good output when the network has been trained with datasets of different sizes. Indeed, having a lower amount of training data would speed up the computational time to train the generator. Another main aim of the paper is to explore the latent space obtained in order to measure (quantitatively and qualitatively) how good this space is to describe the smooth transitions from one syllable to another. This is a way to check if the generator model is able not only to interpolate between syllables but also to generalize (i.e., the latent space is not a collection of homogeneous regions separated by sharp transitions).

Section 4.2 contains an introduction to Generative Adversarial Networks (GANs), a brief mention to the models that are relevant for this study and the description of the architecture of WaveGAN. Section 4.3.1 describes the pre-processing we did to prepare the canary recordings. Section 4.3.2 explains our experimental setup by introducing the training dataset and the different training conditions we used to train WaveGAN. Section 4.3.3 introduces the metrics we used to evaluate the performance of the generator. The tools used to evaluate the model are in Sections 4.3.4 and 4.3.5.

Section 4.4 contains the results obtained in this study. First, Section 4.4.1 shows how the training dataset can be evaluated using our evaluation metrics.

Section 4.4.2 shows the analysis we carried out to evaluate the performance of the generator across time. We observe the generator capacity of covering all the repertoire and the number of syllables not resembling to the real ones (i.e. not recognized as an example of a known class). We first show performance of the generator after one training as a proof of concept, then we compare 10 different instances of the training without changing the experimental setup. The capacity of the generator to reproduce realistic samples is confirmed by a qualitative analysis that we carried out at the end of one instance of the training. Section 4.4.3 shows some examples of generated data and how the output space (i.e. the space of the generated syllables) can be explored and covered by the generator.

Section 4.4.4 and Section 4.4.5 show the same type of analysis presented in Section 4.4.2 applied to compare the generator performances when using different conditions for the latent space dimension, or for the size of the training dataset.

Section 4.5 summarizes the results we obtained and explains the advantages and the limitations of the network. Moreover, we discuss how a generator model such the one investigated by this study could be used in future work within the frame of vocal learning models.

4.2 GAN background

Recently, several generative models have been proposed, such as Wavenet (Oord et al., 2016a), Variational autoencoders (VAEs) (Kingma and Welling, 2013) and Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). In particular, GANs have been introduced for the first time by Goodfellow et al. (2014) as a novel class of machine learning frameworks. Two models, the generator model and the discriminator model, compete to become better at their objective. The generator model aims to be able to produce samples that come from the training data distribution, without having access to any information regarding them. On the contrary, the discriminator has access to the distribution of the training dataset, and should be able to discriminate between a sample which belongs to the training dataset and a sample generated by the generator model (Goodfellow, 2016). In the original formulation, the Jensen-Shannon divergence is used as a loss function. To improve the performance of GANs, several authors proposed new GANs varying the model architecture or the loss function. For example, a more stable training has been obtained using deep convolutional neural networks for both the generator and the discriminator: this is the case for Deep Convolutional GAN (DCGAN) proposed by Radford et al. (DCGAN) (Radford et al., 2015). Alternatively, Berthelot et al. (2017) proposed Boundary Equilibrium GAN (BEGAN) where the discriminator is an auto-encoder (Zhao et al., 2016). Arjovsky et al. used a DCGAN architecture in Wasserstein GAN (WGAN) (Arjovsky et al., 2017) and improved it by modifying the loss

function definition, such that it is robust to changes in the network architecture. Firstly, the role of the discriminator has changed: it is not making anymore a direct choice to assess whether a sample is real or fake. Instead, the discriminator acts as a critic and provides the generator a loss allowing it to be trained until it reaches an optimum. An advantage of the critic is that it can't saturate and converges to a linear function, whereas the classic discriminator could learn too quickly, becoming not reliable (Arjovsky et al., 2017). Secondly, the loss is based on the computation of Wasserstein distance, which gives stability to the GAN and avoids mode collapsing (Dong et al., 2018). Moreover, weight clipping is used to enforce the continuity of the loss function. Finally, WGAN has been improved by the introduction of a regularization parameter, usually called λ. Gulrajani et al. (Gulrajani et al., 2017) proposed the addition of a Gradient Penalty (GP) term in the loss function, driven by λ. GP penalizes any deviation of the gradient norm of the critic. This enables a faster training and less parameter tuning. This model is called WGAN with Gradient Penalty (WGAN-GP). Alternatively, Petzka et al. (2017) proposed the addition of a Lipshitz Penalty (LP) term in the loss function, driven by λ. LP penalized only larger (> 1) deviations of the gradient norm (and not all of them, as GP does). For small values of λ WGAN-GP and WGAN-LP perform similarly, whereas for larger values of λ the performance of WGAN-GP are more λ-dependent (Petzka et al., 2017).

MuseGAN (Dong et al., 2018) and WaveGAN (Donahue et al., 2018) are two examples of application of WGAN-GP to achieve sound generation. Dong et al. (2018) used a dataset of multi-track piano-rolls derived from the Lakh Midi Dataset (LMD) (Raffel, 2016) and train WGAN-GP to generate multi-track sequences. Donahue et al. (2018) trained WGAN-GP to generate music (piano, drums), speech (i.e. the Speech Commands Zero through Nine-SC09 dataset) and, to a lesser extent, birdsong (using a training dataset composed by wild recordings of different species).

Donahue et al. (2018) highlighted using Principal Component Analysis (PCA) how images and waveforms are different in their structure. Principal components capture gradient, intensity and characteristics of the edges for images, whereas principal components

form a periodic basis that decompose the audio signal for waveforms (Donahue et al., 2018). Indeed, audio signals show more periodicity than images. Thus, a larger receptive field to process audio signals is introduced in WaveGAN. This is similar to what Oord et al. (2016a) did in WaveNet, where dilated convolutions has been used to increase the effective receptive field of the model. WaveGAN architecture is based on the architecture of DCGAN (Radford et al., 2015) which uses GANs for image synthesis. DCGAN (Radford et al., 2015) generator is a CNN where transposed convolution is used to upsample low-resolution feature maps into a high-resolution image. As for DCGAN, the generator and the discriminator of WaveGAN are CNNs. Donahue et al. (2018) used WGAN-GP (Gulrajani et al., 2017) strategy during training. This strategy consists in the introduction of a gradient penalty term in the loss function, driven by the regularization hyperparameter λ. The advantages of using GP are a faster training and the avoidance of problems coming from weight clipping (used to force the gradient to stay below a certain threshold), which were present in the original formulation (Gulrajani et al., 2017). Please refer to Appendix 4.2 for further details about the architecture of WaveGAN.

4.3 Methods

4.3.1 Data pre-processing

Canaries sing sequences of syllables organized in phrases: a phrase is a short part of the song where the same syllable is repeated a certain number of times (Markowitz et al., 2013). The starting dataset was composed of a repertoire of 27 classes of syllables organized in labeled phrases (i.e. each phrase was already assigned to a specific class), manually sorted from X hours of recording at $44100Hz$ from an adult canary. First, we focus on the selection of the syllables. We considered a subset of the repertoire. Among the 27 classes, we focused on 16 classes in order to have only classes with enough samples: $A, B1, B2, C, D, E, H, J1, J2, L, M, N, O, Q, R, V$. We performed the following steps on each phrase:

1. We downsampled each phrase to a sampling rate of $16000Hz$.

2. From each phrase we made a first syllable selection tuning three parameters: the amplitude threshold, the minimal duration of the syllable and the duration of the gap (i.e., the silence between two consecutive syllables).

3. We applied to all the selected syllables a high-pass filter of order 5 to remove all frequencies below $700Hz$.

Figure 4.1 shows a summary of the selection procedure for syllable $J2$. The upper panel shows the downsampled phrase. The lower panel highlights each syllable of the phrase: for each syllable, the green dashed line represents the onset (i.e., the beginning of the syllable), and the blue dashed line represents the offset (i.e., the end of the syllable).

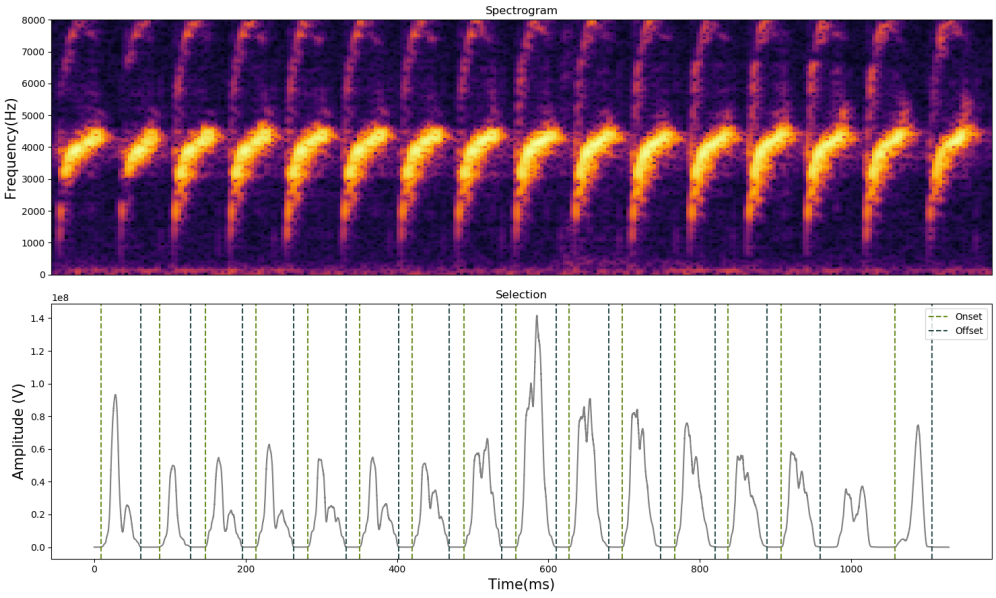


Figure 4.1: **Selection of syllable $J2$.** The upper panel shows the spectrogram of an example phrase of syllable $J2$. The lower panel shows the selected syllables: the green vertical lines represent the onset of each syllable, the blue vertical lines represent the offset of each syllable. We used onset and offset to determine where each syllable begins and ends.

The automatic selection based on an amplitude threshold, the duration of the syllable and the duration of the gap, could fail to select correctly the syllable. We performed a

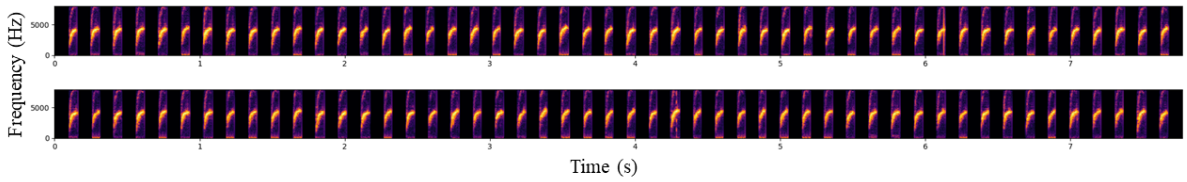


Figure 4.2: **Example of single syllables $J2$.** 100 random selected syllables from the totality of the phrases belonging to class $J2$. To select the syllables we used three parameters: amplitude threshold, minimal duration of the syllable and duration of the gap between two consecutive syllables. Other examples of selected syllables form other classes can be found in Appendix 4.1

visual inspection after selection on duration to filter misclassified and miscut elements. The resulting dataset contains 72155 syllables from the 16 classes. Figure 4.2 shows an example of a phrase from class $J2$ and 100 examples of samples that have been selected from class $J2$, randomly selected. See Appendix 4.1 for a detailed explanation of the syllable selection procedure.

4.3.2 Experimental setup

We selected a balanced training dataset in order to consider almost the same number of samples per syllable type. Among all the selected syllables, we used a subset of $16k$ syllables: $1k$ syllables per class. From now on, we will refer to this balanced dataset as the training dataset. Each syllable has been padded with silence to obtain recordings having a length of exactly 1 s. This step has been done to create a dataset resembling the speech dataset that was originally used to train WaveGAN (Donahue et al., 2018). We used the balanced training dataset to train both the classifier (described in Section 4.3.4) and the network (described in Section 4.2).

We used the original WaveGAN (Donahue et al., 2018) setup to train the network. We used the original network architecture with gradient penalty option, with $\lambda = 10$ and Adam optimizer in the training phase. We used a batch size of 64 samples and we trained the discriminator 5 times more than the generator.

We tested different conditions for the latent space dimension, varying it from $ld = 6$

to $ld = 1$. In the following, we will refer to one of these particular conditions using, for instance, 6-dimensional WaveGAN to refer to a GAN trained with a 6-dimensional latent space, or 5-dimensional WaveGAN to refer to a GAN trained with a 5-dimensional latent space. We tested different sizes for the training dataset. For this analysis, we used the preliminary training dataset described in Appendix 4.1: first, we trained WaveGAN using the complete dataset, then we reduced it by a factor of ~ 8 and finally by a factor of ~ 16.

We first trained WaveGAN using a preliminary dataset (see Appendix 4.1). We trained 3 instances per latent space condition (keeping the dataset size fixed) and dataset size condition (keeping the latent space dimension fixed), to observe the performance of the generator across time. Then, we used the optimal combination of latent space dimension and dataset size to train 10 instances of WaveGAN with the training dataset introduced at the beginning of this Section. For all the instances, we trained the network until epoch ~ 1000, saving the model every ~ 15 epochs. After training, we used the generator to produce new syllables at every saved epoch. We evaluated the generated data using both a quantitative and a qualitative measure, as described in Sections 4.3.3 and 4.3.3.

4.3.3 Evaluation

We evaluated the performance of the generator across epochs and across training conditions, in order to see the optimal choice of parameters. As explained in Section 4.3.2, after training, we used the generator to produce new syllables every 15 epochs and we used the classifier described in Section 4.3.4 to identify the generated syllables as elements of one class of the vocabulary. To this aim, accordingly with the type of evaluation we want to perform, we defined two vocabularies and trained two classifiers.

First, we used the balanced training dataset to train a classifier able to provide the probability of each sample belonging to each class of a vocabulary composed by the 16 classes of the repertoire. We refer to this model as *classifier-REAL*. Then, we introduced the possibility for the classifier to classify a sample as not belonging to any of the 16

classes of the repertoire, but to an alternative unknown class. For simplicity of notation, we will call the set of unknown classes X. We define a class $x \in X$ as a class representing either white noise, or samples containing a lot of noise and with a low variability one from the other (usually, resembling early generations), or to a class of samples resembling the real ones but followed by some artifacts (usually, such samples start to happen at late stages of training). We trained a classifier able to provide the probability of each syllable belonging to 21 classes: the 16 classes of the repertoire, a white noise (WN) class, an overtraining (OT) class, and three $EARLY$ classes, respectively obtained from epochs 15, 30 and 45. To train the classifier to recognize the alternative unknown classes $x \in X$, in addition to the usual training dataset, we used three additional sets of generated samples. In summary, to train the classifier we used:

- *EARLY 15*: $1k$ samples of early generations, obtained after $\sim$ 15 epochs using two different instances of a 3-dimensional WaveGAN (respectively, 500 samples per instance);
- *EARLY 30*: $1k$ samples of early generations, obtained after $\sim$ 30 epochs using two different instances of a 3-dimensional WaveGAN (respectively, 500 samples per instance);
- *EARLY 45*: samples of early generations, obtained after $\sim$ 45 epochs using two different instances of a 3-dimensional WaveGAN (respectively, 500 samples per instance);
- OT: $1k$ samples obtained when two instances of a 3-dimensional WaveGAN reach overtraining (respectively, 500 samples per instance);
- WN: $1k$ samples of artificial white noise.

The two different instances used to define the classes above are instances *Ex 0* and *Ex 1* in Figure 4.8. We will refer to this classifier as *classifier-EXT* (where EXT stands for extended).

Quantitative evaluation

A quantitative measure is a measure which allows to understand if the model is able to reproduce a wide enough variety of samples (Borji, 2019), and provides a preliminary measure of whether or not the generated data resemble the real syllables. To observe the generator's performance across time we will describe how many classes it is able to produce and how many syllables per class it is able to produce in average. In this way, we studied the stability of WaveGAN across different instances of training. Moreover, we computed the Inception Score (IS) at several epochs of the training, we observed its evolution and we compared it with the IS obtained from the training dataset. IS has been proposed by Salimans et al. (2016) as a quantitative measure to evaluate GANs: a pre-trained deep learning neural network model for image classification provides the probability of each image belonging to each class. This information is then summarized in the IS, which is defined as follow:

$$IS = exp(\mathbb{E}(KL(p(y|x)||p(y)))), \tag{4.1}$$

where KL stands for Kullback-Leibler divergence. Given a problem with N classes, $IS \in [1, N]$. IS provides both a measure of the quality and of the entropy of the generations, giving an idea of the generator ability to produce a wide set of new data (Salimans et al., 2016). The IS as a method of objective evaluation for GANs performance has been used by several authors (Donahue et al., 2018; Gulrajani et al., 2017).

Qualitative evaluation

First, we based our qualitative analysis on spectrogram analysis. On the one hand, we computed the mean spectrogram of the generated data, and we compared it with the mean spectrogram of the dataset and with the repertoire. To obtain the mean spectrogram, for each class, we first aligned the syllables' envelope, then we computed the mean of the spectrograms of all the syllables belonging to that class. On the other hand, we observed the spatial organization of the data using Uniform Manifold Approximation and

Projection for Dimension Reduction (UMAP) (McInnes et al., 2018). We applied UMAP to the spectrograms of the samples. Further details about the algorithm can be found in Section 4.3.5.

Secondly, we explored the latent space to study the continuity of the generations. We generated syllables for each small variation of the latent vector.

- **One component variation.** We selected a random latent vector $z \sim R^3([-1,1])$ to generate a baseline syllable using a 3-dimensional WaveGAN after training. Then, we moved one by one the components of the vector by a variation step equal to $v_{step} = 0.05$. To explore critical points (i.e., where a bigger variation arises a sudden non-smooth change between two syllables) we moved one by one the components of the vector by a variation step equal to $v_{step} = 0.01$ and $v_{step} = 0.001$. Moreover, we used a step $v_{step} = 0.001$ to vary one by one each component of z between $[-1,1]$ and we compared the generations obtained by each variation (one per component of the latent vector) with a set of $16k$ generated data from the same epoch.

- **Three components variation.** We first selected 2 syllables s_1 and s_2 and their correspondent latent vectors z_1 and z_2, where $z_1, z_2 \in R^3([-1,1])$. Then, we moved in the latent space from z_1 to z_2 using a variable step depending on the distance between the components of z_1 and z_2. That is, the step applied to each component i is

$$step[i] = \begin{cases} \frac{z_1[i]-z_2[i]}{N_{step}} & z_1[i] > z_2[i], \\ -\frac{|z_1[i]-z_2[i]|}{N_{step}} & otherwise, \end{cases} \tag{4.2}$$

  where N_{steps} is the number of steps. We used $N_{steps} = 1000$. We used *classifier-EXT* to identify the syllables and see which class is assigned to the transition between s_1 and s_2. We used the UMAP representation to compare the variations with a bigger set of generated data.

Finally, we used human judgment to provide an additional qualitative evaluation. We asked 2 people to participate in a syllable recognition test organized as described below.

- **Training phase.** Recognition of a sample of 100 syllables from the training dataset: for the first 50 (or less, if not needed anymore) the person is authorized to look at the repertoire to classify the syllables. After each guess, the person can also look at the correct answer to learn and become more specialized. Then, the last 50 syllables have to be recognized. No helps is allowed here, and no correct answer can be seen. In the training phase, the available classes are the 16 classes of the repertoire.

- **Testing phase.** Recognition of a sample of 200 syllables generated after training, without the possibility of consulting the repertoire. In the testing phase, the available classes are the 16 classes of the repertoire plus a general X class which, ideally, represents the alternative unknown classes (three $EARLY$ classes, OT class and WN class) recognized by the classifier.

Then, we computed the proportion of agreement without considering the chance agreement (i.e., that percentage of agreement that would have occurred anyway) using Cohen's kappa coefficient (Cohen, 1960; Artstein and Poesio, 2008). If $\kappa_{Cohen} = 0$, then the agreement is equal to agreement by chance. Alternatively, $\kappa_{Cohen} \in (0, 1]$ represents a positive agreement and $\kappa_{Cohen} = 1$ represents a perfect agreement between two judges (Cohen, 1960). We computed κ_{Cohen} for each of the participant with respect to *classifier-EXT* and for each couple of participants.

4.3.4 Classifier

The two classifiers (*classifier-REAL* and *classifier-EXT*) used during the evaluation phase (described in Section 4.3.3) are *Echo State Networks* (ESNs) (Jaeger, 2001), a type of artificial neural network. ESNs and Support Vector Machines (SVMs) work similarly, by embedding input data into a high dimensional space using a random non linear transformation. However, unlike SVMs, ESNs are designed to manipulate sequential data, like recurrent neural networks (RNNs), and are relevant candidates for a sound classification task.

The classifiers were fed with mel-frequency cepstral coefficients (MFCCs) representations of the syllables, a low dimensional spectral representation of sound. We extracted 20 MFCCs features per time step, one time step being defined as the result of a spectral analysis window of $64ms$ applied to the $16kHz$ audio signal, with a $32ms$ jump between each time step. Because the generated syllables tend to have higher amplitude than the real ones, we used only the first and the second derivative of the extracted MFCC signals to remove any influence of the signal amplitude in the representations. Otherwise, the amplitude difference would bias the classifiers decisions, as it would be artificially easier to separate real samples from generated samples only by comparing the average power of the signals. First and second derivatives of the MFCC signal are also known to be good representations of vocal signals, as they give relevant clues on the temporal dynamics of the vocalizations.

The ESNs trained on the classification tasks are built using *RerservoirPy* Python toolkit (Trouvain et al., 2020)[1], and are described by the following equations:

$$\begin{aligned} x(n) &= (1-\alpha)x(n-1) \\ &\quad + \alpha \tanh\left(W_{in}u(n) + Wx(n-1)\right) \end{aligned} \tag{4.3}$$

$$y(n) = W_{out}x(n) \tag{4.4}$$

where $u(n)$, $x(n)$ and $y(n)$ are respectively the input features, the internal state of the network and the output vector at time step n. W_{in} stores the connection weights between the inputs and the neuronal units of the network. These weights are sampled from a discrete bi-modal distribution, i.e. are randomly chosen between 1 or -1. The proportion of non-zero connection weights is fixed to 10%. W_{in} is defined in $\mathrm{R}^{N\times I}$, where N is the number of neuronal units and I is the dimension of the input. In our case $I = 41$, with input features being 20 derivatives of MFCCs, 20 second derivatives of MFCCs, and a constant bias equal to 1. Each set of features is scaled by multiplying the corresponding

[1] https://github.com/reservoirpy/reservoirpy

connection weights in W_{in} by a constant. The first derivatives and the bias are scaled by a factor 1, and the second derivatives are scaled by a factor 0.7. W stores the connection weights of the neuronal units of the network. These weights are sampled from a standard normal distribution, with a proportion of non-zero connections between neuronal units fixed to 10%. W is defined in $\mathrm{R}^{N \times N}$, with N equal to 1000. W is scaled using a factor equal to a fixed *spectral radius* of 0.5 divided by the largest absolute eigenvalue of W. The α parameter, called *leaking rate*, is set to 0.05. It controls the time constant of the ESNs and allows information from past internal states to be fed to future internal states. All connection weights defined in W_{in} and W are fixed during the training phase of the ESNs, as opposed to machine learning algorithms using gradient descent. Only the $W_{out} \in \mathrm{R}^{N \times V}$ readout matrix is learned during training, where V is the dimension of the vocabulary used to classify the input features, i.e. $V = 16$ for *classifier-REAL* and $V = 21$ for *classifier-EXT*. The readout weights are learned using a simple linear regression between all the internal states x generated from the inputs and all the expected values of the output y. A *L2* regularization coefficient of value 10^{-8} is applied during the linear regression.

The classifiers then output a vector $\hat{y}(n)$ of dimension V representing its activation for each time step n and for each category of syllable in the vocabulary. Then, these output activities are averaged over the whole sequence of MFCC time steps representing each syllables. A softmax operation finally provides a probability distribution representing the chance for the syllable to belong to one of the classes of the vocabulary.

A preliminary version of the classifier can be found in Appendix 4.3.1 and a further analysis of its robustness can be found in Appendix 4.3.2.

4.3.5 Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP)

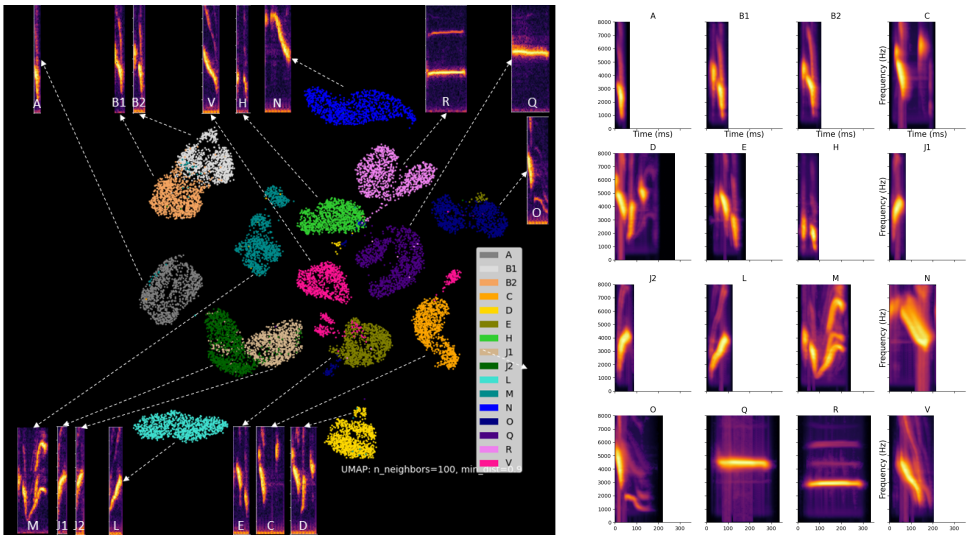
Similarly to t-SNE, Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) (McInnes et al., 2018) is a dimension reduction technique. As a plus, it can be used to perform non-linear dimension reduction, and it has a higher computational

power (i.e., it is faster than other reduction techniques). The axis are not meaningful to identify a discriminant feature (i.e., they do not represent, for example, the pitch of the syllable, or another syllable-related feature). Instead, there are hyperparameters that simply work well to represent the given dataset: for instance, the size of the neighborhood used to estimate the manifold structure of the data and the minimum distance apart that points are allowed to be. The tuning of these hyperparameters, respectively, allows to obtain a more or less local representation of the data (where a higher size of the neighborhood translates in a more local representation) and to pack or not pack together the points in the clusters (where a higher distance allows a more sparse representation). For instance, the power of UMAP to represent birdsongs' data have been explored by Sainburg et al. (2019).

4.4 Results

4.4.1 Analysis of the training dataset

This section describes the training dataset and highlights the performance of our evaluation metrics on it. The majority of the samples belong to 16 independent clusters in the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) (McInnes et al., 2018) representation (left panel of Figure **??**a). Besides, the representation shows similarities between syllables *B1* and *B2* and syllables *J1* ad *J2*: the clusters of syllables *B1* (cream cluster) and *B2* (light orange cluster) and the clusters of syllables *J1* (dark green cluster) ad *J2* (light brown cluster) lie very close. These similarities can be noticed also in the spectrogram representation (Figure **??**b). For this reason, in further representations using UMAP syllables *B1* and *B2*, *J1* ad *J2* have been grouped, respectively, into syllable *B* (keeping the light orange color to represent the cluster) and *J* (keeping the dark green color to represent the cluster) because of their high similarity. Each template shown in Figure 4.3a can be compared with the correspondent mean spectrogram (Figure 4.3b).

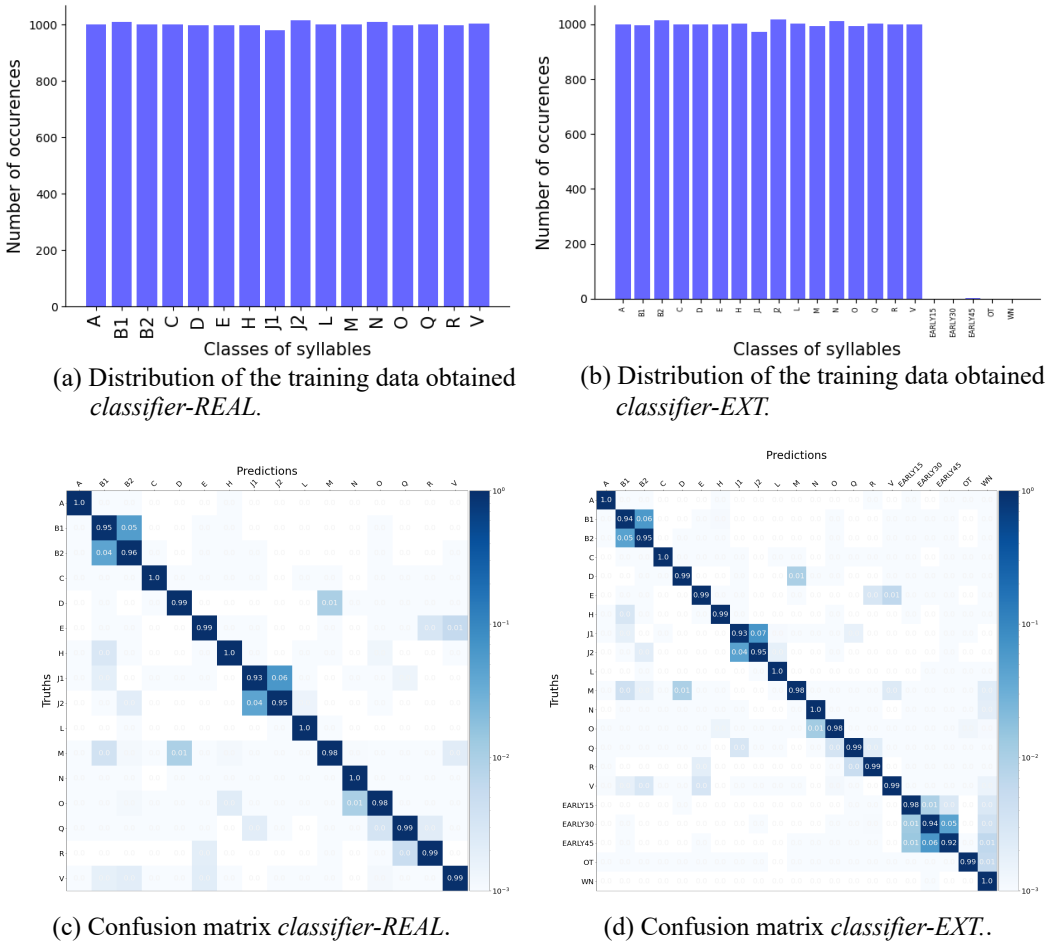


(a) UMAP representation and repertoire.

(b) Mean spectrogram.

Figure 4.3: **Repertoire.** (a) UMAP (McInnes et al., 2018) representation of the training dataset. Each cluster represents a class of the repertoire and a template syllable of each class is highlighted with the corresponding spectrogram (an arrow connects each cluster to the corresponding template). Syllables *B1* (cream cluster) and *B2* (light orange cluster) and syllables *J1* (dark green cluster) and *J2* (light brown cluster) lie very close: this is why in the following UMAP representations we will unify them in two big clusters, syllable *B* (will be cream cluster) and syllable *J* (will be dark green cluster). (b) Mean spectrogram computed after envelope alignment of the waveforms. To obtain the UMAP representation and the mean spectrogram we used $1k$ syllables per class (i.e., the training dataset) and their real labels. No classifier has been applied to assign each syllable to the correct class.

The average number of syllables recognized for each of the 16 classes of the repertoire is close to $1k$ both for *classifier-REAL* and *classifier-EXT* (panels (a-b) of Figure 4.4). This is coherent with the fact that the training dataset contains $1k$ samples per class. Moreover, the alternative unknown classes $x \in X$ (i.e., three classes of *EARLY* generations, the overtraining (OT) class, and the artificial white noise (WN)) identified by *classifier-EXT* contain a negligible number of elements, or zero elements (Figure 4.4b). The level of confidence of the classifier in making the correct assignment can be represented using the confusion matrix relative to the predictions. On the diagonal the optimal condition would be to have values equal to 1, elsewhere the optimal condition would be to have values equal



(a) Distribution of the training data obtained *classifier-REAL*.

(b) Distribution of the training data obtained *classifier-EXT*.

(c) Confusion matrix *classifier-REAL*.

(d) Confusion matrix *classifier-EXT*..

Figure 4.4: **Training data analysis**. Panels (a-b) show the distribution obtained using *classifier-REAL* and *classifier-EXT* on the training dataset. In both cases, the average number of syllables per class is $1k$ for each of the 16 classes of the repertoire. The remaining 5 columns (on the right part) in panel (b) represent the alternative unknown classes $x \in X$ (*EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*). Panels (c-d) show the confusion matrix (i.e. the level of confidence of the classifier in making the correct assignment) relative, respectively, to *classifier-REAL* and *classifier-EXT*. Each square represent the level of confidence of the classifier in making the correct assignment. The level of confidence is expressed on a logarithmic scale using shades of blue.

to zero. Looking at the confusion matrix for both *classifier-REAL* and *classifier-EXT*, a confusion between syllable $B1$ and syllable $B2$, and between syllable $J1$ and syllable $J2$ can be seen on the diagonal in correspondence of these elements (sub-diagonal elements

are darker than the others for these 4 syllables). This confusion can be explained with the fact that phrases formed by syllables $B1$ and phrases formed by syllables $B2$, and phrases formed by syllables $J1$ and phrases formed by syllables $J2$ differ in repetition rate more than in frequency.

The Inception Score (IS) obtained for the training dataset after using *classifier-REAL* to recognize the syllables is $IS_{real} = 15,92$. The range of the IS for our dataset is $IS \in [1, 16]$.

4.4.2 Evaluation of the model

We used the classifier described in Section 4.3.3 to obtain a quantitative analysis. We saved the model every $\sim$ 15 epochs until epoch $\sim$ 1000. In order to determine a good epoch to generate samples resembling the real ones, we are interested in the performance of the generator across time. At the beginning of training (epoch 0 in Figure 4.5), there is no variation between the syllables. At epoch 15 the generations start to be coherent in duration but remain noisy and unclear. The resemblance increases at epoch 45 for some syllables (e.g., syllable E) but remains generally low for most of them. Finally, epoch 984 generations show a shape comparable with the training data in Figure 4.3.

Quantitative evaluation

At each epoch, we generated $1k$ samples and we used *classifier-EXT* to calculate the probability distribution of the classes. Figure 4.6 shows the distribution obtained from the classifier at 4 example epochs: epochs 15, 106, 212, 318, 514 and 984. Each columns represents one of the 21 classes of the vocabulary: the 16 classes of the repertoire and 5 alternative unknown classes $x \in X$ (i.e., three classes of $EARLY$ generations, the overtraining (OT) class, and the artificial white noise (WN)). These results have been obtained from an instance of training where the latent space dimension was fixed at $ld = 3$. It is possible to observe a decrease in the number of syllables belonging to an alternative class, whereas an increasing number of syllables characterize the classes of the repertoire.

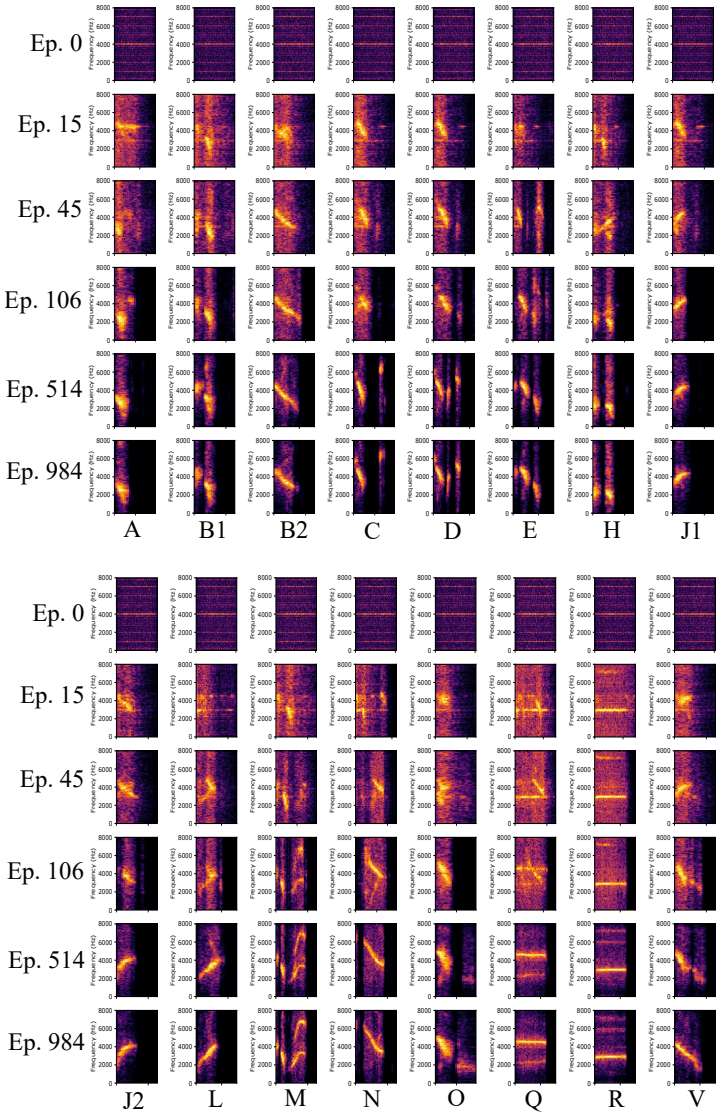


Figure 4.5: **Generations across time.** Example of one selected syllable per class across time. Each syllable has been first generated at epoch 984 and recognized using *classifier-EXT*. Then, the latent vector associated at each syllable has been used to generate the same syllable at epochs 0,15, 45, 106 and 514. At epoch 0 the generations do not vary from one class to another. At epoch 15 the generations start to be coherent in duration but remain noisy and unclear. At epoch 45 some syllables are more distinguishable than at earlier epochs, but the majority remains noisy and indistinguishable. As the training goes on, the generations resemble more and more the real syllables (epochs 106 and 514). The generations obtained at epoch 984 have a clear distinguishable shape. Here, the generator obtained from the instance *Ex 6* in Figure 4.7 has been used to generate the syllables across time; latent space dimension $ld = 3$; *Ep.* stands for epoch.

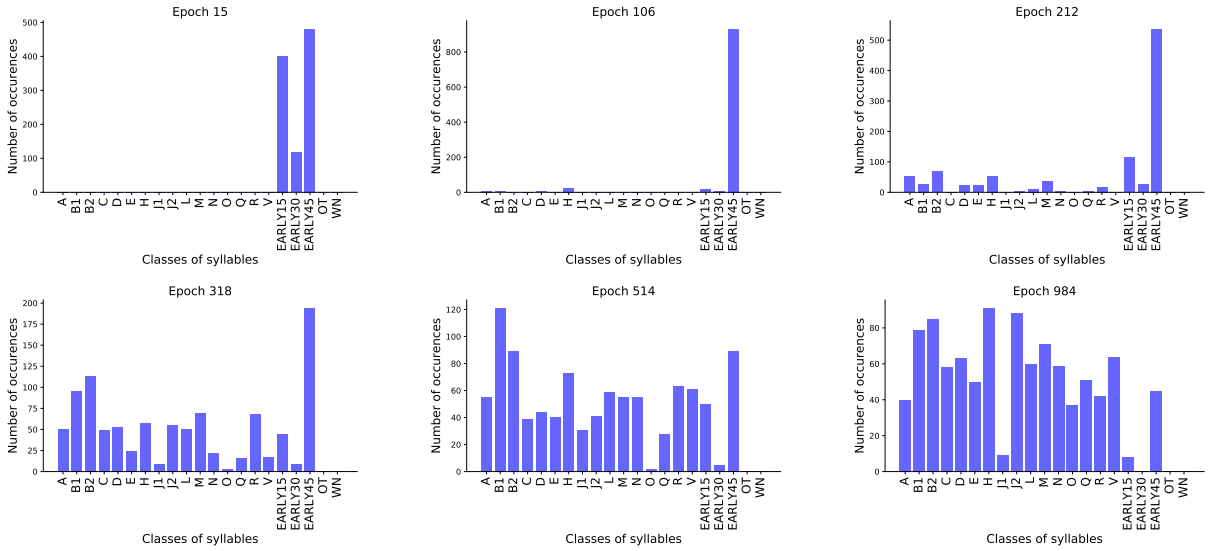
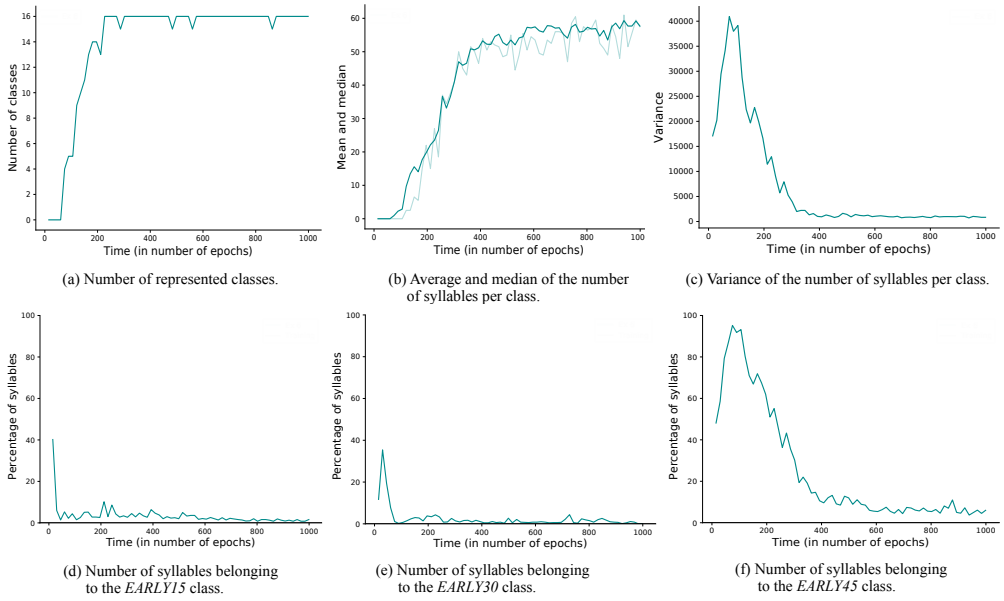


Figure 4.6: **Distribution of the classes**. Distribution of $1k$ generated samples after 15, 106, 212, 318, 514, and 984 epochs of training. Each column represents a class of the vocabulary: a syllable from the repertoire, or an alternative unknown class $x \in X$ (*EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*). The number of syllables belonging to a class $x \in X$ decreases over time, whereas the average number of syllables belonging to the classes of the repertoire increases. The latent space dimension is $ld = 3$ and *classifier-EXT* has been used to classify the generated data.

The capacity of a 3-dimensional generator model to produce samples that are classified as belonging to the repertoire increases over time. After $\sim$ 200 epochs of training, the generator is able to cover all the syllables of the repertoire (Figure 4.7a). In parallel, the total variance (considering both the classes of the repertoire and the 5 alternative unknown classes) decreases over time (Figure 4.7c). That is, at early stages of training the majority of the samples generated are classified by *classifier-EXT* as elements of an alternative unknown class $x \in X$ (Figures 4.7d, 4.7e and 4.7f). Then, the number of samples recognized by *classifier-EXT* as belonging to a class of the repertoire increases leading to an increasing in the average number of syllables per class (Figure 4.7b). The dark green line shows the evolution of the mean, whereas the light green line shows the evolution of the median. After epoch $\sim$ 600 the average number of syllables per class remains stable.



(a) Number of represented classes.

(b) Average and median of the number of syllables per class.

(c) Variance of the number of syllables per class.

(d) Number of syllables belonging to the *EARLY15* class.

(e) Number of syllables belonging to the *EARLY30* class.

(f) Number of syllables belonging to the *EARLY45* class.

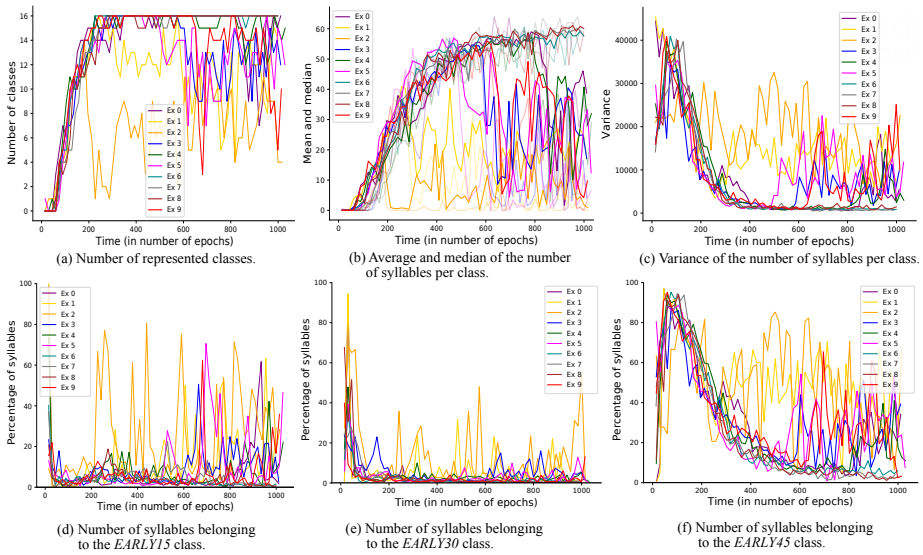
Figure 4.7: **Analysis of a 3-dimensional generator using *classifier-EXT***. Statistical analysis performed on the classifier distribution of one instance of the training. Panel (a) shows the number of classes represented by the generated data: that is, at each epoch, how many syllables of the repertoire are covered by the generator. The number of syllable reaches the maximum value 16 relatively quick and then stays high during all the training. Panel (b) shows the average number (dark colored lines) of elements per class and the median (light colored lines): across time, with some instability, we can observe an increasing of the mean across time. To compute the quantities in panels (a) and (b), alternative unknown classes $x \in X$ have not been taken into account. Panel (c) shows the evolution of the variance of how many syllable per class have been produced. Here, $x \in X$ classes are included. The variance starts at a high value when the majority of the samples produced are not classified as syllables of the repertoire, then it decreases when the generator becomes better at producing syllables. Panels (d-f) show the percentage of syllables that are classified as belonging to one of the alternative classes $x \in X$: from the left to the right, classes *EARLY15*, *EARLY30* and *EARLY45*.

At the same time, the number of syllables belonging to one of the alternative classes $x \in X$ (*EARLY15*, *EARLY30* and *EARLY45*) decreases over time (Figures 4.7d, 4.7e and 4.7f). Classes *EARLY15* and *EARLY30* decrease faster with respect to *EARLY45*, which, eventually, never reaches a level comparable to the percentage found in the training data. Although this show that the generator produces better and better samples over time,

further analysis are needed to understand if the quality of the samples keep increasing after epoch 600.

The concept of overtraining arises comparing 10 different instances of training of 3-dimensional WaveGANs. Not all the instances show the capability of the generator of producing syllables belonging to all the classes of the repertoire (Figure 4.8a). Instances *Ex 2* (orange line), *Ex 1* (yellow line) and *Ex 5* (magenta line) show an early drop and, eventually, never reach to cover the repertoire. Nevertheless, the other instances show a drop at an advanced stage of the training (i.e., after epoch 600) or never drop. Similarly, the instances showing instability in Figure 4.8a show instability in the average number of syllables recognized per class (Figure 4.8b) and in the variance of the number of syllables recognized per class (Figure 4.8c). Moreover, alternative unknown classes, and in particular class *EARLY45*, are more represented even in advanced epochs of the training for those instances showing instability, as shown in Figures 4.8(d-f). Such an instability might characteriz the beginning of overtraining: the generator starts to produce samples that are recognized as elements of an *EARLY* class or of class *OT*. The latter is more represented in instances showing overtraining (Figure 4.9). After, the generator is not able to recover. Further analysis are needed to understand how to carefully describe overtraining.

Finally, as a last quantitative measure of the generator performance, we computed the Inception Score (IS) across time. For 3 instances (i.e., *Ex 0*, *Ex 2* and *Ex 6*) we generated $16k$ samples every ~ 100 epochs and we used *classifier-REAL* to classify them. Then, we computed the IS as in Equation 4.1 at each saved epoch. Figure 4.10 shows the IS across time with respect to IS_{train} (black line). The IS of *Ex 0* and *Ex 6* increases across time and, eventually, they reach a maximum value around 13 (for reference, the IS obtained from the training dataset is $IS_{train} = 15,92$); contrarily, the IS of *Ex 2* increases at the beginning but remains low during all the training. The IS of *Ex 0* suddenly breaks around epoch 800: this behavior confirms the results shown in Figure 4.8.



(a) Number of represented classes.
(b) Average and median of the number of syllables per class.
(c) Variance of the number of syllables per class.
(d) Number of syllables belonging to the *EARLY15* class.
(e) Number of syllables belonging to the *EARLY30* class.
(f) Number of syllables belonging to the *EARLY45* class.

Figure 4.8: **Different instances of a 3-dimensional WaveGAN**. Statistical analysis performed on the classifier distribution of 10 instances of training. Panel (a) shows the number of classes represented by the generated data: that is, at each epoch, how many syllables of the repertoire are covered by the generator. An early drop (i.e., before epoch 600) in the number of represented classes can be seen for three instances (i.e, *Ex 2* – orange line, *Ex 1* – yellow line and *Ex 5* – magenta line). Panel (b) shows the average number (dark colored lines) of elements per class and the median (light colored lines): depending on the instance, the mean and the median could increase over time and remain stable (*Ex 6* – light blue line, *Ex 7* – gray line, *Ex. 8* – burgundy line), or increase until a certain epoch and then drop (*Ex 3*-blue line, *Ex 4* – green line, *Ex 5* – magenta lines, *Ex 9*-red line), or remain low for all the duration of the training (*Ex 1* – yellow line, *Ex 2* – orange line). To compute the quantities in panels (a) and (b), alternative unknown classes $x \in X$ have not been taken into account. Panel (c) shows the evolution of the variance of how many syllable per class have been produced. Here, $x \in X$ classes are included. For successful instances of training (*Ex 6* – light blue line, *Ex 7* – gray line, *Ex. 8* – burgundy line), the variance starts at a high value when the majority of the samples produced are not classified as syllables of the repertoire, then it decreases when the generator becomes better at producing syllables. Eventually, it increases again later (*Ex 3* – blue line, *Ex 4* – green line, *Ex 5* – magenta line, *Ex 9* – red line) or never decrease enough (*Ex 1* – yellow line, *Ex 2* – orange line). Panels (d-f) show the percentage of syllables that are classified as belonging to one of the alternative classes $x \in X$: from the left to the right, classes *EARLY15*, *EARLY30* and *EARLY45*. Alternative unknown classes, and in particular class *EARLY45*, are more present even in advanced epochs of the training for those instances where we identify overtraining (all but *Ex. 6* – light blue line, *Ex 7* – gray line and *Ex. 8* – burgundy line.

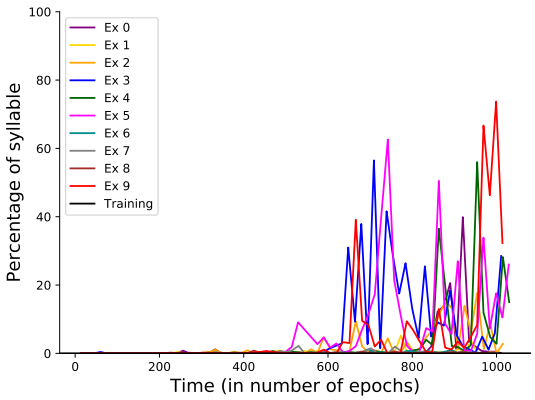


Figure 4.9: **Overtraining**. Comparison between 10 instances of training: percentage of syllables classified by *classifier-EXT* as elements of class OT (i.e., overtraining). Instances *Ex 0* (purple line), *Ex 1* (yellow line), *Ex 2* (orange line), *Ex 3* (blue line), *Ex 4* (green line), *Ex 5* (magenta line), *Ex 9* (red line) show an increasing number of syllables classified as belonging to class OT. These instances show instability also in the average an variance of the syllables belonging to the classes of the repertoire and in the number of syllables belonging to an $EARLY$ class (see Figure 4.8). Latent space dimension: $ld = 3$.

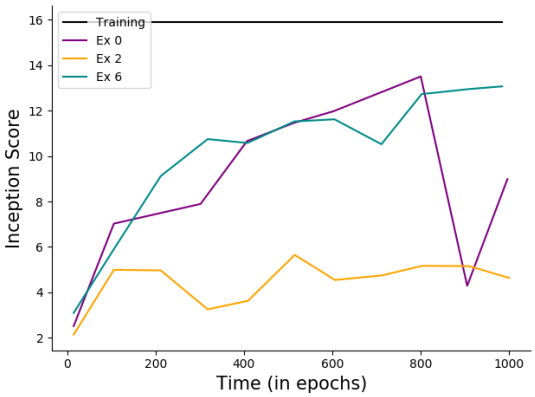


Figure 4.10: **Inception Score across time.** IS relative to the generator of instances *Ex 0* (violet line), *Ex 2* (orange line) and *Ex 6* (light blue line, used as baseline) with respect to IS_{train} (black line). IS for the instances *Ex 0* and *Ex 6* increases over time, whereas IS for the instance *Ex 2* remains low over time. Here, for each instance we generated $16k$ samples every ~ 100 epochs and we used *classifier-REAL* to classify them. This figure is not complete: we will add the IS relative to the remaining instances of training before the defense. The generator of each instance has been used to produce $16k$ syllables across time and *classifier-EXT* has been used to identify the generated syllables.

Table 4.1 summarizes the IS of the training dataset and each of the selected instances: we considered the highest IS within the time range shown in Figure 4.10.

Table 4.1: **Inception Score**. IS of the training dataset and the selected instances: we considered the highest IS within the time range shown in Figure 4.10. For our dataset, $IS \in [1, 16]$.

Dataset	Epoch	IS
Training	-	15,92
Ex 0	800	13,51
Ex 2	515	5,65
Ex 6 (baseline)	984	13,07

Qualitative evaluation

For qualitative evaluation, we focused on instance *Ex 6* of training. We chose to focus on this instance because it remains stable until epoch $\sim$ 1000 (see Figure 4.7) and we obtained an increasing and high Inception Score (see Figure 4.10 and Table 4.1). Indeed, *Ex 6* is also the example we used since the beginning of Section 4.4.2 as a baseline example.

The generator produces a higher number of good syllables over time. At early epochs of training, when not all the classes are covered by the generator (empty boxes in Figure 4.11(a-b)), the mean spectrograms are blurry and show syllables difficult to recognize as belonging to a class of the repertoire (see Figure 4.3). The spectrograms look often as a mix of syllables coming from several classes (e.g, syllable *A* or syllable *R* in Figures 4.11(a-b)). At epoch 514 (Figure 4.11c) all the syllables can be produced by the generator and only a few syllables remain difficult to be recognized as belonging to a class of the repertoire (e.g., syllable *B2* and syllable *L*). Nevertheless, a lot of syllables are clearly recognizable (e.g., syllable *N* and syllable *Q*). Finally, at advanced stages of training, all the spectrograms show a recognizable syllable.

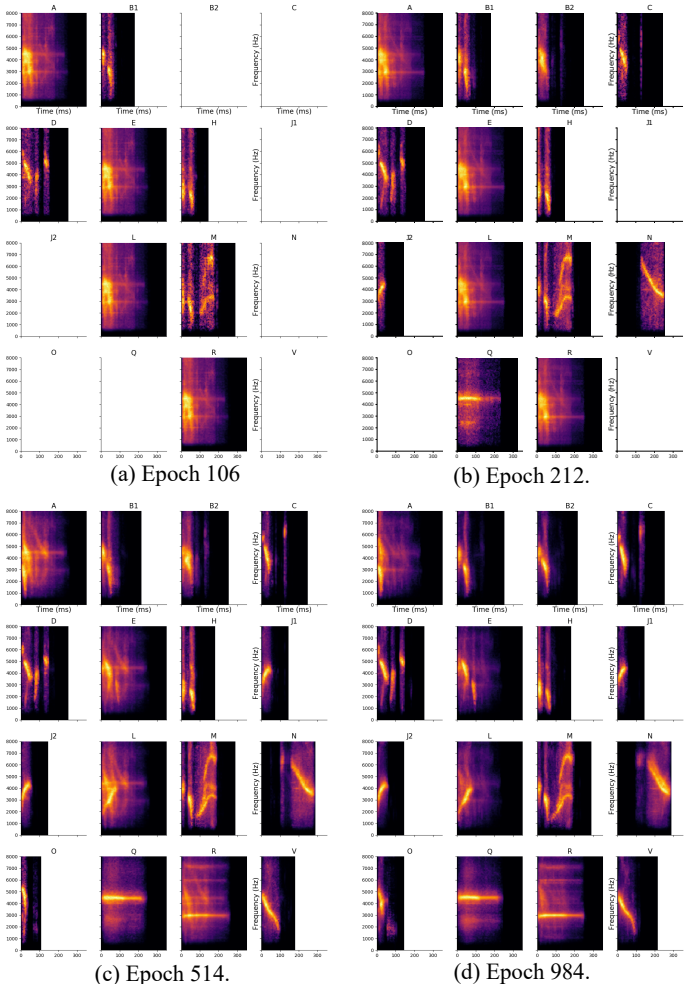
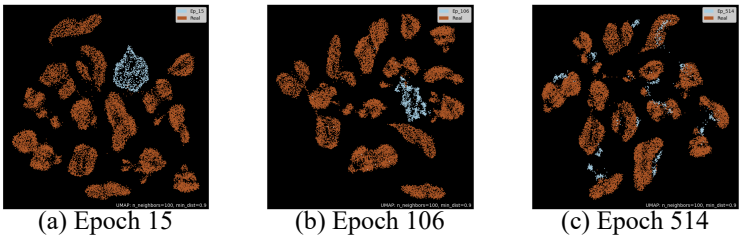
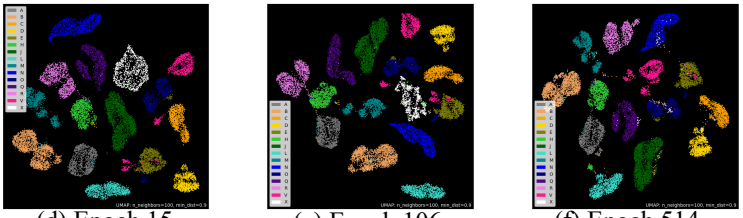


Figure 4.11: **Mean spectrogram across time.** Mean spectrogram of $1k$ the syllables generated at epoch 106 (a), 212 (b) 514 (c) and 984 (d). Empty boxes in panels (a) and (b) mean that at epoch 106 and 212 not all the repertoire can be covered by the generator and no syllables have been recognized by *classifier-EXT* as elements of the not represented classes (e.g., class *B2*). At epochs 106 (panel (a)) and 212 (panel (a)) the correct duration and, eventually, the content of the syllables can be grasped. At epoch 514 (panel (c)) almost all the syllables can be distinguished and only a few remain noisy and unclear (*A*, *E*, *L*, *R*). At epoch 984 (panel (d)) all the syllables are clear and distinguishable. They can be compared with the repertoire in Figure 4.3. Here, the training has been done using $ld = 3$, the generator obtained from the instance *Ex 6* in Figure 4.7 has been used to generate the syllables across time and *classifier-EXT* has been used to identify the generated syllables.

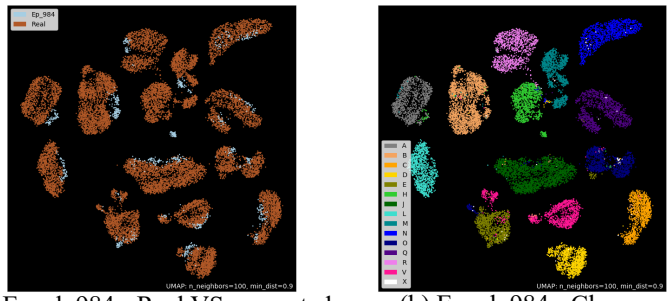
The UMAP representation of the generated data and the training data show that



(a) Epoch 15 (b) Epoch 106 (c) Epoch 514



(d) Epoch 15 (e) Epoch 106 (f) Epoch 514



(g) Epoch 984 - Real VS generated (h) Epoch 984 - Classe

Figure 4.12: **Syllable space representation across time.** Syllable space representation obtained from the training dataset ($16k$ syllables) and $1k$ syllables generated at epochs 15, 106, 514 and 984 using UMAP McInnes et al. (2018). Panels (a-c) and (g) show the training data (brown points) and the generated data (blue points). These four figures are different because the analyzed dataset differs for the $1k$ generations specific of each epoch. Panels (d-f) and (h) show the same representation of panels (a-c) and (g) with the classes of syllables visible. Each cluster/color corresponds to one class of the repertoire and class X (in white) represents the cumulative class of the alternative unknown classes (in this case, *EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*). Here, the training has been done using $ld = 3$, the generator of instance *Ex 6* has been used to generate the syllables and *classifier-EXT* has been used to identify both the generated data and the training data.

(1) the generated data are grouped together as they were an additional cluster with respect to the ones obtained from the training data (Figures 4.12(a-c, g)) and (2) the generated data belong to the same cluster of the training data (Figures 4.12(d-f, h)).

On the one hand, the generated data spread over time, moving from being a cluster in itself (light blue points in Figures 4.12(a-c, g)) to taking a conformation compatible with the training data (brown points in Figures 4.12(a-c, g)). On the other hand, the generated data are mostly constituted by syllables belonging to class X (the cumulative class of the alternative unknown classes (in this case, *EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*) at early stages of training (Figures 4.12(d-e)). Later, the majority of the generated data belongs to the same class as the closer cluster of training data in the UMAP representation (Figure 4.12h). Further observations regarding the comparison between the UMAP representation of training and generated data are discussed in Figure 4.38 in Appendix 4.4.

The UMAP representation of the generated data shows smoother transitions from one cluster to another (Figure 4.13). These transitions can be either represented by points belonging to one of the classes of the repertoire (e.g., between J and C) or to class X (e.g., between N and Q). This observation highlights an interesting perspective for which elements classified in class X can be seen as intermediate elements between two syllables. We will deal with this concept in Section 4.4.3.

Finally, human judgment confirms the goodness of the classifier and of the generated data. Each judge evaluated the same set of 200 samples generated at epoch 984 using the generator of instance *Ex 6*. The judges had the possibility to classify each of them as an element of one of the 16 classes of the repertoire or as an element of an alternative unknown class $x \in X$. Table 4.2 contains the Cohen's kappa values across the human judges and versus *classifier-EXT*.

For further observations about the stability of the training, Appendix 4.4.2 shows how, combining the UMAP representation and the mean spectrogram, it is possible to compare three consecutive epochs of training to check the stability of the generator.

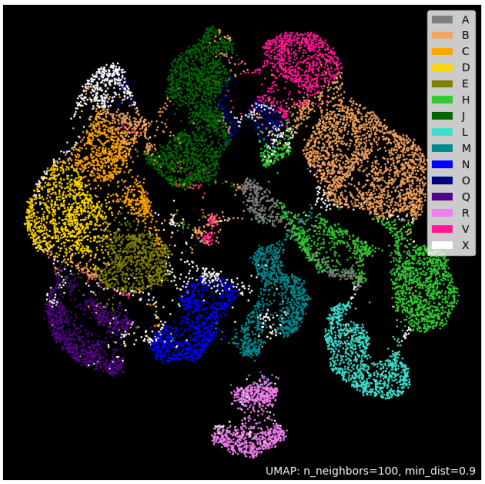


Figure 4.13: **UMAP representation of the generated data.** UMAP representation of $16k$ generated syllables. Each cluster/color correspond to one class of the repertoire and class X (in white) represents the cumulative class of the alternative unknown classes (in this case, *EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*). Here, the training has been done using $ld = 3$, the generator obtained from the instance *Ex 6* in Figure 4.7 has been used to generate the syllables at epoch 984 and *classifier-EXT* has been used to identify the generated syllables.

4.4.3 Latent space exploration

To explore the continuity of the latent space we used the generator obtained from one instance of training of a 3-dimensional WaveGAN. In particular, we used *Ex 6* at epoch 984. We used two different strategy to (1) explore the latent space in all the direction starting from one point (one component variation described in Section 4.3.3) and (2) observe the transition from one syllable to another (three components variation described in Section 4.3.3).

First, we selected a random latent vector $z \sim R^3([-1, 1])$ to create a baseline syllable, then continuously move from one representation to another by changing one dimension of the latent space by a fix variation step. The syllables highlighted by the red squares in Figure 4.14 (i.e., first *C* and *J2*, then *V* and *V*) are an example of non-smooth transitions. For these particular transitions, we considered the two consecutive syllables obtained from

Table 4.2: **Human Judgment**. Cohen's kappa coefficient computed per each couple across human judges and versus *classifier-EXT*. Each judge evaluated 200 syllables produced using the generator of instance *Ex 6*. The kappa coefficients κ_{Cohen} obtained across judges and with respect to *classifier-EXT* are comparable.

	Cohen's kappa
Judge 1 vs Judge 2	$0,74$
Judge 2 vs *classifier-EXT*	$0,79$
Judge 1 vs *classifier-EXT*	$0,73$

the first variation step (i.e., two consecutive syllables contained in two consecutive red squares) and we applied a smaller variation step. The bottom-left panel of Figure 4.14 shows the latter operation applied to the first non-smooth transition in the first component and highlights a new non-smooth transition that needs to be investigated. Using the same procedure, we applied a smaller variation step and reached a point at which we cannot see any more clear evidence of non-smooth transitions (bottom-right panel of Figure 4.14). The investigation of the second non-smooth transition highlighted in the upper panel of Figure 4.14 and additional examples from the the second and the third components can be found in the supplementary material.

A better visualization of such smooth transition can be observed in the UMAP representation shown in Figure 4.15. The exploration obtained varying, one by one, each component of a random latent vector $z \sim R^3([-1,1])$ by a step equal to $v_{step} = 0.001$ shows smooth passages between one syllable to another. It is not surprising that a difficulty in the classification arises between syllable V (magenta cluster) and syllable C (orange cluster): indeed, syllable V is a syllable that shares its content with other syllables of the repertoire, causing uncertainty and errors for the classifier. The UMAP representation of the training dataset (Figure 4.3a) and then the UMAP representation of the generated data (Figure 4.13) already show multiple cluster locations for syllable V.

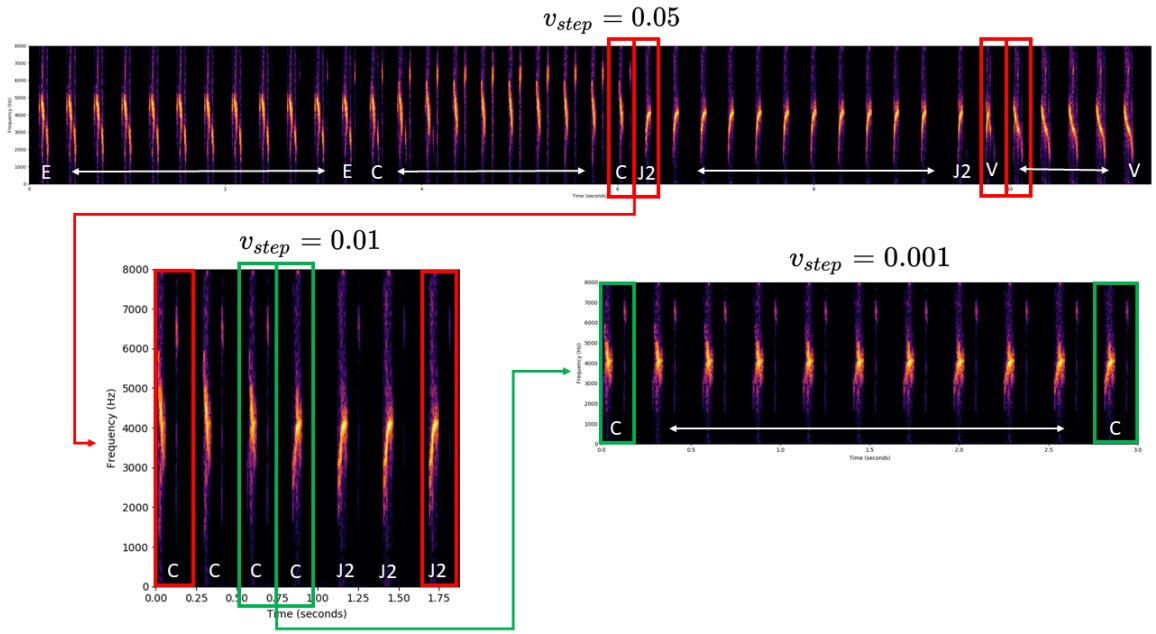
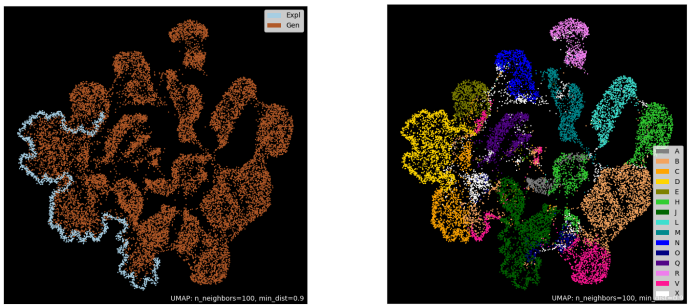
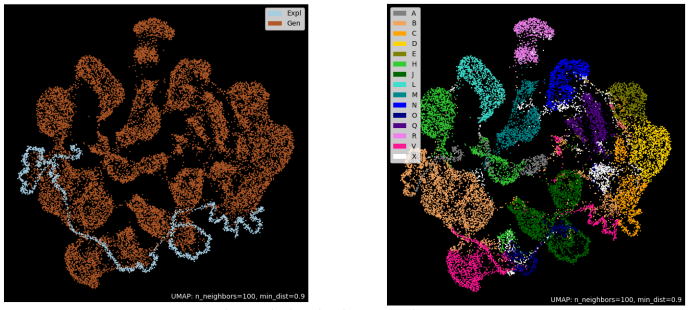


Figure 4.14: **Exploration of the latent space: one component variation. First component.**. We selected a random latent vector $z \sim R^3([-1, 1])$ to create a baseline syllable. Then, we moved one by one the components of the vector by a variation step equal to $v_{step} = 0.05$. We observed all the syllables produced to look at how they evolve and if there are non-smooth transitions. The upper panel shows the exploration of the first component of the latent vector obtained with $v_{step} = 0.05$. The syllables highlighted by the red squares (i.e., first *C* and *J2*, then *V* and *V*) are an example of non-smooth transitions. For these particular transitions, we considered the two consecutive syllables obtained from the first variation step (i.e., two consecutive syllables contained in two consecutive red squares) and we applied a variation step of $v_{step} = 0.01$ to the first component of the latent vector to generate intermediate syllables. The bottom-left panel shows the latter operation applied to the first non-smooth transition in the first component and highlights a new non-smooth transition that needs to be investigated. To do so, as shown in the bottom-right panel, we used a variation step equal to $v_{step} = 0.001$ to generate intermediate syllables. We have reached a point at which we cannot see any more clear evidence of non-smooth transitions. The syllables have been produced by the 3-dimensional generator obtained from instance *Ex 6* and the name of the syllable for this analysis has been provided by *classifier-EXT*.



(a) Variation in the first component



(b) Variation in the second component

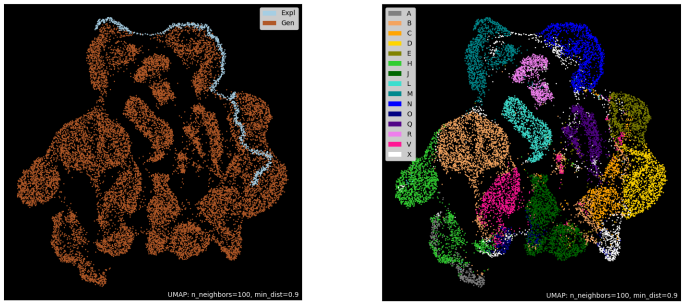
(c) Variation in the third component

Figure 4.15: **Latent vector components variation.** Panel (a) represents the first component variation, panel (b) represents the second component variation and panel (c) represents the third component variation. The left panels show the variational data (generated at each step) (light blue points) and $16k$ generated data from the same model at the same epoch (brown points). Each cluster/color in the right panels corresponds to one class of the repertoire and class X (in white) represents the cumulative class of the alternative unknown classes (in this case, *EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*). Here, the training has been done using $ld = 3$. A latent vector has been randomly selected and a step $v_{step} = 0.001$ has been applied to its component, one by one. The generator obtained from instance *Ex 6* of the training has been used *classifier-EXT* has been used to identify the generated syllables from the generator of instance *Ex 6*.

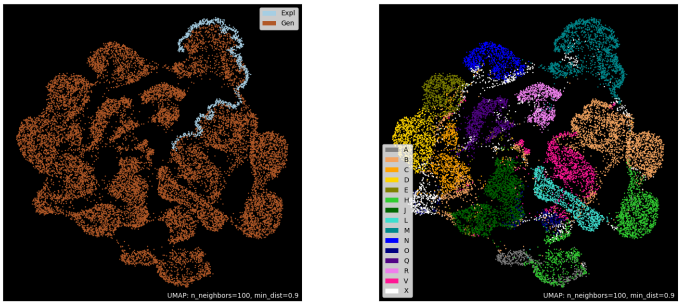
We used $N_{steps} = 1000$ and we computed the three components transition from (i) syllable M and syllable D, (ii) syllable M and syllable V and (iii) syllable H and syllable N. Indeed, these couples of syllables are not adjacent in the UMAP representation of the generated data (see Figure 4.13). For all the transitions, it is possible to observe that the variational data cross the generated data (left panels of Figure 4.16) giving rise to a smooth change of syllable class (right panels of Figure 4.16). Interestingly, the transition between syllable M and syllable D (Figure 4.16a, right panel) shows that the intermediate syllables between two classes are sometimes recognized as class X (white points between syllable M (turquoise cluster) and syllable N (blue cluster). Moreover, the transition between syllable H and syllabe N shows (1) an interesting stretch of syllable B (cream cluster) connecting syllable B and syllable Q and (2) the uncertainty of the classifier in differentiating between syllable H (light green cluster) and syllable A (gray cluster). The latter was already shown by the UMAP representation of the generated data (Figure 4.13). This results, combined with the UMAP representation we obtained for the generated data and shown in Figure 4.13, highlight the fact that the generator is not only producing samples from the training dataset but also other samples. This allows to move realistically between two syllables, as shown in Figure 4.16.

4.4.4 Latent space dimension

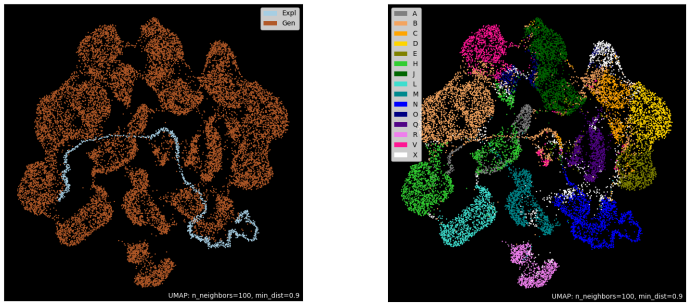
Until now, we have shown results obtained from a 3-dimensional WaveGAN. Indeed, a 3-dimensional WaveGAN works nicely and as good as higher dimensional WaveGANs and, at the same time, lower dimensional WaveGANs do not show good performances or stability (Figure 4.17). Although all conditions allows the generator to reach the ability of producing all the type of syllables, 1-dimensional and 2-dimensional WaveGANs converge later with respect to higher dimensional WaveGANs (Figure 4.17a). At the same time, 1-dimensional and 2-dimensional WaveGANs show a slower increase in the average number of syllables belonging to each class of the repertoire (Figure 4.17b) and a slower decrease in the variance of the number of syllables per class. Nevertheless, as already mentioned



(a) Transition from M to D



(b) Transition from M to V

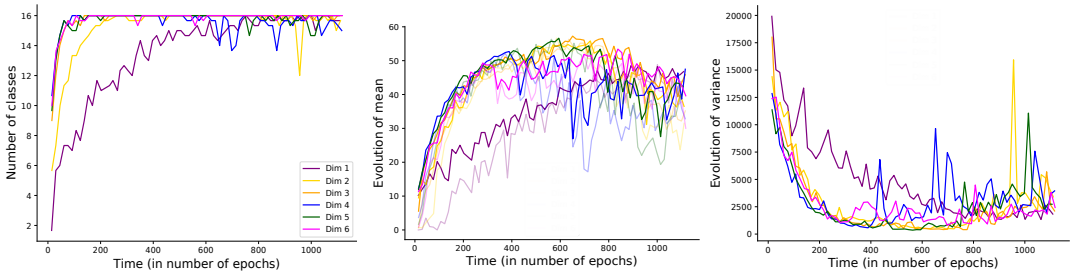


(c) Transition from H to N

Figure 4.16: **Transition between two generated syllables.** Panel (a) represents the transition between M (turquoise cluster) and syllable D (yellow cluster), panel (b) the transition between M (turquoise cluster) and syllable V (magenta cluster) and panel (c) the transition between H (light green cluster) and syllable N (blue cluster). The left panels show the variational data (generated at each step) (light blue points) and $16k$ generated data from the same model at the same epoch (brown points). Each cluster/color in the right panels corresponds to one class of the repertoire and class X (in white) represents the cumulative class of the alternative unknown classes (in this case, *EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*). Here, the training has been done using $ld = 3$ and *classifier-EXT* has been used to identify the generated syllables from the generator of instance *Ex 6*.

in Section 4.4.2, towards the end of the training a generalized instability could appear (what we refer to as overtraining), without distinction of latent space dimension (e.g., overtraining happens for $ld = 3$ as well as for $ld = 6$). This can be seen in a decrease in the mean and median evolution (Figure 4.17b) and in an increase in the variance evolution (Figure 4.17c).

A qualitative measure of the generated syllables obtained from WaveGANs of different dimensions (e.g., for different conditions of the latent space dimension) can be found in Figure 4.45 in Appendix 4.4.4.



(a) Number of represented classes.

(b) Average and median of the number of syllables per class.

(c) Variance of the number of syllables per class.

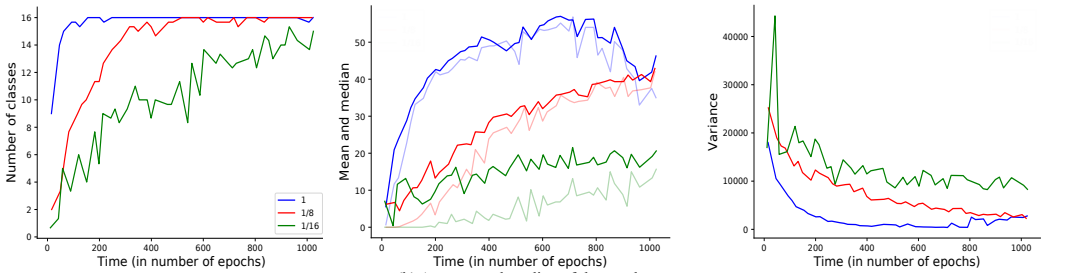
Figure 4.17: **Comparison between different latent space dimensions**. Each line represents the average over 3 instances of training at a particular latent space dimension. Panel (a) shows how many syllables of the repertoire are covered by the generator across time. Panel (b) shows how many syllable per class have been generated in average. The dark lines show the evolution of the mean, whereas the light lines shows the evolution of the median. To build these two panels (a) and (b), only the repertoire's classes have been taken into account. Panel (c) shows the evolution of the variance of how many syllable per class have been produced. A 3-dimensional WaveGAN (orange line) reaches convergence as good as higher-dimensional WaveGANs (blue, green and magenta lines) and better than lower dimensional WaveGANs (purple and yellow lines). We varied the latent space dimension as $ld = 1, 2, 3, 4, 5, 6$ and we kept the training dataset fixed. The training of all the instances of WaveGAN has been done using the preliminary training dataset introduced in Appendix 4.1 and *classifier-PRE* (see Appendix 4.3.1) has been used to identify the syllables. A complete version of this figure is available in Appendix 4.4.4.

4.4.5 Training dataset dimension

A larger dataset allows WaveGAN to converge faster and better (Figure 4.18). At the beginning of the training the generator was only able to produce a limited number of syllable types, whereas it becomes able to reproduce all of them after some epochs of training (Figure 4.18a). In particular, for both trainings with *dataset 1* and *dataset 8*, after $\sim$ 250 epochs, the generator is already able to produce samples which the classifier can recognize as one element of the repertoire. Likewise, Figure 4.18b shows how the average number of syllables generated per class across time increases more when WaveGAN is trained with a larger dataset (blue lines) than when WaveGAN is trained with smaller datasets (red and green lines). Similarly, the variance evolution shows a faster and better convergence when WaveGAN is trained with a larger dataset (blue lines) than when WaveGAN is trained with smaller datasets (red and green lines). All the trainings show a decreasing behavior, with *dataset 1* (blue line) showing a higher slope at the beginning and reaching a lower minimum. Moreover, *dataset 1* (blue lines) shows instability in the average number of syllable belonging to each class of the repertoire and a slight increase in variance around epoch 850: this could be identified as the beginning of what we refer to as overtraining.

4.5 Discussion

In this paper we analyzed the ability of WaveGAN (Donahue et al., 2018) to produce canary syllables in the case of a low dimensional latent space and of a limited training dataset size. In particular, we studied the capability of the generator of producing good quality outputs similar to the training outputs. We first used a RNN-based classifier to recognize the generated syllables. Then we evaluated them quantitatively and qualitatively. On the one hand, we looked at the statistical properties of a set of generated data (e.g., average and median number of syllables produced per class, variance, inception score). On the other hand, we used the mean spectrogram and the UMAP (McInnes



(a) Number of represented classes.

(b) Average and median of the number of syllables per class.

(c) Variance of the number of elements per class.

Figure 4.18: **Comparison between datasets of a different size**. We varied the dataset size as $d = 23456, 3600, 1600$ and we kept fix the latent space dimension at $ld = 3$. Each line in this figure represents the average over 3 instances of training at a particular dataset size condition. Panel (a) shows how many syllables of the repertoire are covered by the generator across time. Panel (b) shows how many syllable per class have been generated in average. The dark lines show the evolution of the mean, whereas the light lines shows the evolution of the median. To build these two panels (a) and (b), only the repertoire's classes have been taken into account. Panel (c) shows the evolution of the variance of how many syllable per class have been produced. A dataset of bigger size (blue line) allows better and faster convergence than having a dataset of lower sizes (red and green lines). The training of all the instances of WaveGAN has been done using the preliminary training dataset introduced in Appendix 4.1 and *classifier-PRE* (see Appendix 4.3.1) has been used to identify the syllables. A complete version of this figure is available in Appendix 4.4.4.

et al., 2018) representations (1) to compare the generated data with the training data and (2) to study the stability of the training. We explored the latent space to highlight its smoothness: we first explored how small changes in the latent space influence the generations (see Figure 4.14); then, we explored the latent space by moving from one syllable to another (see Figure 4.16). In summary, we concluded that a low-dimensional GAN is able to generalize and interpolate between syllables, as well as to produce good quality syllables.

The analysis we have we have conducted leads to the conclusion that GANs producing sounds can be used in vocal learning models to model the motor function as an alternative to other sound synthesis. Indeed, motor control in vocal learning models has been often modeled using Ordinary Differential Equations (ODEs) (Amador et al., 2013;

Gardner et al., 2001; Westerman and Miranda, 2002; Maeda, 1989). Such models are usually based on the anatomical structure of the vocal tract and the respiratory apparatus, and, especially for humans, can include a large number of parameters. Originally, WaveGAN (Donahue et al., 2018) has been trained with a 100-dimensional latent space. Notably, we showed that WaveGAN can produce good syllables even if the latent space dimension is reduced to $ld = 3$. This is one of the key points in order to have a biologically motivated vocal sensorimotor model. Indeed, a high-dimensional motor space in a vocal learning model would lead to unrealistic computational time (Pagliarini et al., 2018a). Consequently, we believe that our 3-dimensional WaveGAN could serve as motor function in such sensorimotor model.

Although we believe in the possibility that generative models can be used as motor control in vocal learning models, many aspects of GANs need to be deeply understood. The problem of instability (i.e., like often in deep learning applications, GAN instances do not always reach an optimal performance), the hyperparameter sensibility and the limitations related to overfitting arise the necessity of a high computational time to explore the performance of GANs. For instance, a limitation of WaveGAN (Donahue et al., 2018) is the absence of a stopping criteria for the training: this introduces an incertitude about how to evaluate the learning and its stability (i.e., in number of epochs rather than depending on the loss function value). Moreover, the algorithm proposed is based on the gradient penalty (GP) factor (introduced by (Gulrajani et al., 2017)) but alternative regularization techniques have been proposed (e.g., the Lipshitz penalty (LP) proposed by (Petzka et al., 2017)). These improvements are out of the scope of this paper.

Despite the fact that it is not clear how to evaluate generative models, and especially GANs (Goodfellow, 2016; Borji, 2019), generally both a quantitative and a qualitative measure are used to this end. A quantitative evaluation aims to determine whether or not the generator is producing only examples on which the classifier was trained or always the same example per type. At the same time, it could lack in representing well human perception (Donahue et al., 2018). The latter can be solved by introducing a qualitative measure (e.g., a measure based on human judgment (Borji, 2019)). A qualitative mea-

sure is indispensable to truly understand the quality of the produced samples and their comparability to real recordings in term of whether or not they are comprehensible to an external expert judge. At the same time, the drawbacks of a qualitative measure is that it could be biased (by the experience the human is having), expensive, not efficient in detecting overfitting. Also, a qualitative measure presupposes the interpretability of the data by humans, which is not always possible (Borji, 2019).

Inception Score (IS) (Salimans et al., 2016) is a broadly used quantitative measure which gives an idea about whether or not the generator model is able to produce a wide enough variety of samples. Nevertheless, IS is not able to detect if the generator is producing only examples that belong to the training dataset or if the generator enter mode collapse (i.e., it produces always the same example per type) (Gui et al., 2020). Thus, IS should not be used as a holistic method to evaluate the performance of a model and must be combined with another evaluation method (Barratt and Sharma, 2018). Instead, an additional quantitative measure could be used to have a better understanding of the quality of the training. For instance, Donahue et al. (2018) measured the Euclidian distance in the space of log-Mel spectrograms (1) within the training and the generated data (to measure the intra-dataset diversity with respect to the training data), and (2) between the training and the generated data (to show the ability of WaveGAN of producing sounds not belonging to the training dataset).

The mean and median of the number of syllables produced per class and the IS continuously increase over time and eventually drop if overtraining begins. We call overtraining the fact that, at a certain epoch, a drop in the IS (see Figure 4.10) or in the statistical properties of the classifier distribution (e.g., in the average number of elements per class in panel (b) of Figure 4.8) is shown. However, the concept of overtraining remains unclear. We tried to characterize it using a specific class (OT) which truly arises at advanced stages of training, but never becomes the most represented class (see Figure 4.9). Instead, after the training drops, there is an exponential increase in the number of element assigned to early classes (see Figure 4.8(d-f)).

A qualitative evaluation based on human judgment has been proposed by Salimans

et al. (2016) and Denton et al. (2015) to evaluate GANs trained on MNIST or CIFAR-10 (image datasets). Similarly, Donahue et al. (2018) used human judgment to evaluate WaveGAN performance when training on SC09 (speech dataset). We rely first on the mean spectrogram obtained from the generated data over time (see Figure **??**) to observe at once how the average quality of syllables increases over time, becoming more and more comparable with the repertoire (see Figure 4.3). Then, we rely on the UMAP (McInnes et al., 2018) representation of the spectrograms to compare the generated data with the training data. On the one hand, the generated data belong to the same cluster as the training data (see Figure 4.12) if represented together. On the other hand, when observing the generated data alone (see Figure 4.13), generated data form clusters as the training data do, and a continuity across clusters arises. The power of using UMAP representation to reduce the complexity of songbirds spectrograms has first been shown by Sainburg et al. (2019): an exhaustive set of representations show how UMAP helps in representing not only the repertoire of a single species but also different species at the same time. The result is a two-dimensional representation where the two variables are not features-related but rather two hyperparameters that enables a synthetic, clear and simple representation of the original manifold. In the case of data not familiar for non-expert humans (such as canary syllables), one could design a behavioral protocol to test the accuracy by measuring how canaries perceive the generated data. Although such an experiment could be very interesting, it is complicated to settle and require an expertise outside of the machine learning domain.

Additionally, to understand whether or not the generator is able to generalize and produce samples not belonging to the training dataset we explored the latent space. We analyzed (1) how a small change in the latent space affects the generated syllable and (2) the transition between two different generated syllables. The training dataset is a discrete set, as shown in Figure 4.3, whereas the clusters obtained from a set of generated data appears less discontinuous when there is the transition between one class and another (see Figure 4.13). By moving from one syllable to another in the latent space by doing *realistic* steps, it is possible to move by continuous steps in the UMAP

representation of the spectrograms. Moreover, if the applied step for moving in the latent space is too big (i.e., it allows non-smooth transitions), reducing the step allows to obtain such smooth transitions (see Figure 4.14). As a perspective to this study, one could explore the possibility of enlarging the training dataset by including (1) more classes of syllables, (2) more birds (in terms of number), (3) recordings from different species or (4) recordings from juveniles. These experiments could help understanding syllable generation in songbirds and the limitations of our model. In principle, the addition of more syllables or recordings from different canaries should not arise particular complications for WaveGAN, even if it would introduce complexity in the training dataset and it would increase the computational time (i.e., the number of epochs needed to obtain a generator able to output good syllables). On the contrary, the attempt of using the model on other species is not trivial. For instance, although zebra finches represent an important model for vocal learning, they produce a not purely harmonic song which could not be the ideal training dataset for WaveGAN. Moreover, with respect to canaries, zebra finches exhibit a more complex spectrogram, which might be difficult to categorize for a human judge, due to a poorer perceived visual quality of the spectrograms.

Donahue et al. (2018) trained WaveGAN on a set of wild recordings from different bird species: the generated samples were noisy due to the variability of the dataset without enough recordings for each specie. By introducing a dataset of recordings from a single adult canary, we simplified the complexity of the training dataset and obtained qualitatively better results. The fact that we deal with single syllables simplifies the analysis of the performance of the generator and the exploration of the latent space. If the generator model could produce sequences of syllables it would be difficult to perform the qualitative analysis we propose here (e.g., the mean spectrograms or the transition from one syllable to another in the latent space). Moreover, a model able to generate single syllables may be an adequate tool to generate full canary songs. Such model would require (1) an estimation of the distribution of the delay within syllables of the same class and (2) a probabilistic model estimating (i) how many time a syllable is repeated and (ii) the upcoming syllable in the phrase. Indeed, canary songs are composed of sequences

of phrases, which are themselves repetitions of the same syllable with smooth transitions. The number of syllable repetitions in one phrase is variable. Thus, a GAN trained to produce bouts of a few seconds would not be able to reproduce the variability of canary song with much flexibility. On the contrary, a GAN like the one proposed, which is able to produce smooth controllable transitions between syllables, is likely to be a good tool to generate full canary songs.

Training WaveGAN on both juveniles and adult data is an interesting and rather straightforward extension of this work. For instance, it can be useful to design a motor control function that could produce any possible canary sounds, from juveniles to adults. This would be useful in computational experiments in order to model the sound productions at different stages of learning. It could also be useful for behavioral experiments with birds. A similar experiment has been proposed by Sainburg et al. (2019): generated samples from a Variational Autoencoder (VAE) has been played to a group of European starlings in a decision making experiment. The birds can learn the task with a high proficiency, and the neural responses obtained from electrophysiology vary smoothly when small variations are applied to the stimulus.

Acknowledgment

We would like to thank Catherine Del Negro, Aurore Cazala and Juliette Giraudon for the recording and trascription of the canary data. We also thank Inria for the CORDI-S PhD fellowship grant and LabEx BRAIN for the PhD extension grant.

4.1 Appendix I: Syllable Selection

4.1.1 Preliminary training dataset

We started from the same canary dataset as described in Section 4.3.1. We performed the following steps on each phrase.

1. We downsampled each phrase to a sampling rate of $16000Hz$.
2. From each phrase we made a first syllable selection tuning three parameters: the amplitude threshold, the minimal duration of the syllable and the duration of the gap (i.e., the silence between two consecutive syllables).
3. We added silence after each sample up to a duration of $1s$.

That is, we did not filter the syllables and we did not remove the errors as we did for the dataset described in Section 4.3.1. The results is a dataset with errors more or less evident depending on the syllable class. We did this choice to be able to start training the network and the classifier and see preliminary results while investigating the syllable selection.

Semi-automatic error detection

As mentioned in Section 4.3.1, we manually check the syllables after the automatic selection. In this section we explain which errors could arise from the automatic selection and how we solved them.

First, our algorithm could have failed in cutting a recording because of the presence of a too short gap between syllables. In this case we obtained samples containing more than one syllables. Alternatively, some samples could be too short, which means they contain only a part of the syllable. Finally, a bad initial classification (e.g. a phrase of class A was wrongly classified as class M) could lead either to well-selected syllables belonging to the wrong class, or, again, to a not effective cut. Applying a semi-automatic procedure for syllable selection, including errors, we were able to select 78827 syllables

from the 16 classes. Some syllables (such as syllables belonging to class O, N, or C) are more difficult to select. Often the automatic selection based on amplitude threshold, duration of the syllable and of the gap, fails to select completely the syllable. That is, the beginning or the end of the syllable is systematically not recognized. In this case, we tried to correct manually the selection by adding a certain amount of samples after (or before) the automatic selection: in this way, more syllables can be correctly isolated.

As a preliminary solution for these errors, we applied a filter on the duration of the samples to eliminate samples that are too short (usually, we do not consider syllables shorter than $10ms$) or too long (usually, we do not consider syllables longer than $300ms$). Of course, as for the selection parameters, these threshold values could change depending on the syllable type. This procedure allows to eliminate error due to a cutting failure and resulting in samples containing more than one syllable, samples containing a very short sound (which is not always a syllable), or, occasionally, the wrong, misclassified, syllable. Using a threshold based on the duration, we could remove 6672 errors. That is, we had an error equal to $8,46\%$ after the semi-automatic selection. The clean dataset contains 72155 syllables from the 16 classes. A random set of 100 single syllables (selected from the phrases) belonging to each class of the repertoire are collected in Figure 4.19 to Figure 4.33.

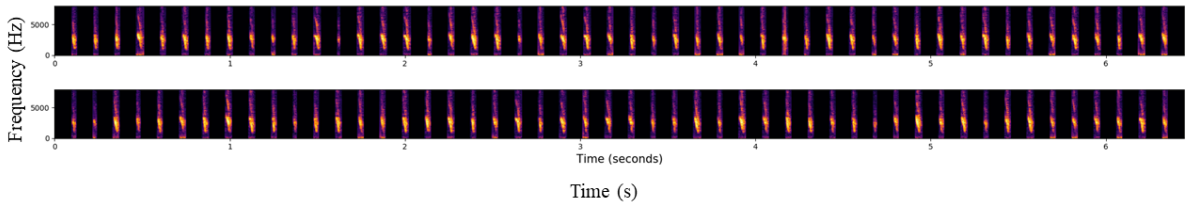


Figure 4.19: **Example of single syllables A.** 100 random selected syllables from the totality of the phrases belonging to class A.

Errors due to an a priori misclassified phrase can't always be solved applying a threshold based on the syllable duration. On the one hand, if the difference between two syllables is evident, the error can be corrected simply using a threshold based on the syllable duration. This is the case of a phrase M wrongly classified as phrase A. The mean duration

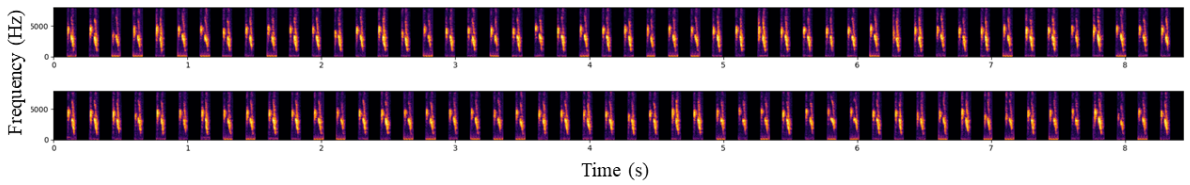


Figure 4.20: **Example of single syllables** $B1$**.** 100 random selected syllables from the totality of the phrases belonging to class $B1$.

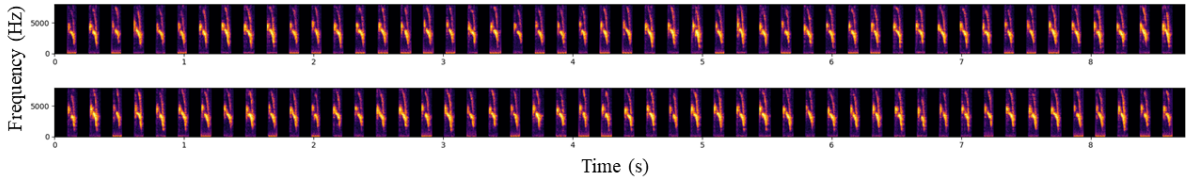


Figure 4.21: **Example of single syllables** $B2$**.** 100 random selected syllables from the totality of the phrases belonging to class $B2$.

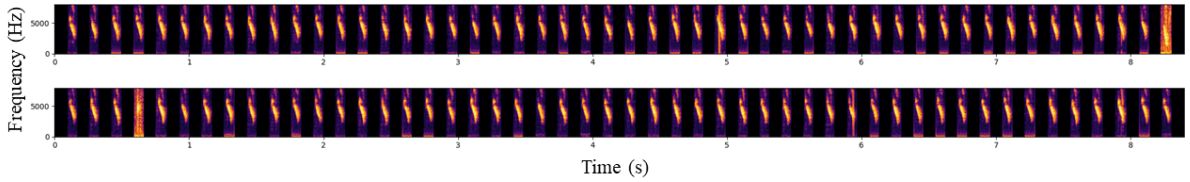


Figure 4.22: **Example of single syllables** C**.** 100 random selected syllables from the totality of the phrases belonging to class C.

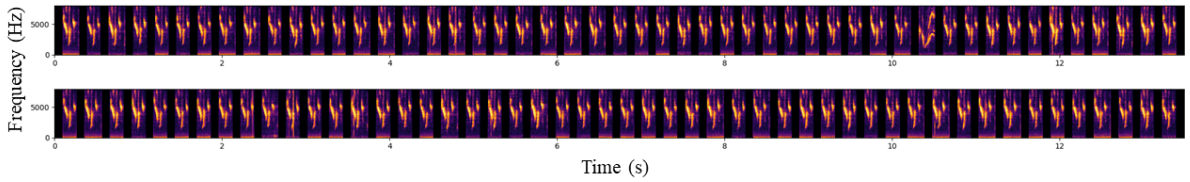


Figure 4.23: **Example of single syllables** D**.** 100 random selected syllables from the totality of the phrases belonging to class D.

of syllable A is much smaller than the mean duration of syllable M: this means that we can fairly assume that a $100ms$ sample can't be syllable A (which has a shorter average duration). Viceversa, we can assume that a $30ms$ sample does not belong to class M. On the other hand, if two syllables share their duration distribution, it becomes more

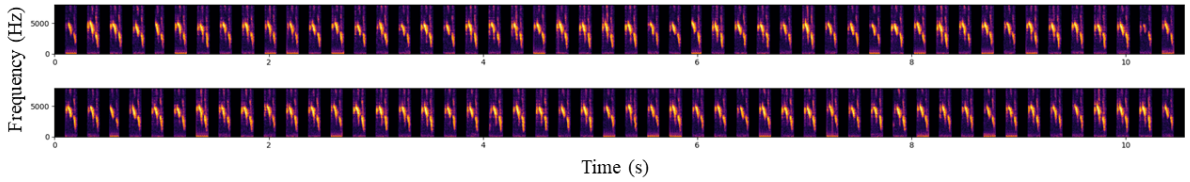


Figure 4.24: **Example of single syllables E.** 100 random selected syllables from the totality of the phrases belonging to class E.

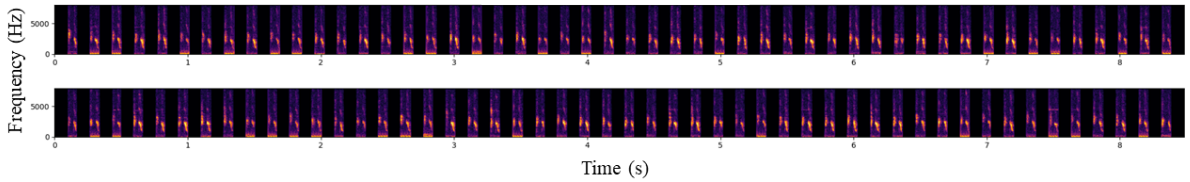


Figure 4.25: **Example of single syllables H.** 100 random selected syllables from the totality of the phrases belonging to class H.

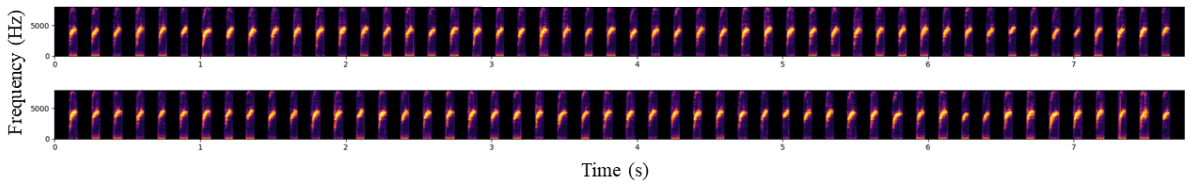


Figure 4.26: **Example of single syllables $J1$.** 100 random selected syllables from the totality of the phrases belonging to class $J1$.

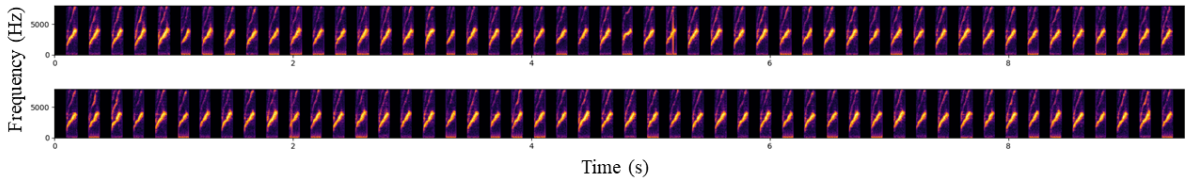


Figure 4.27: **Example of single syllables L.** 100 random selected syllables from the totality of the phrases belonging to class L.

difficult to get rid of samples coming from a wrongly classified phrases. This is the case of a phrase M wrongly classified as phrase D: the two syllables have not only a similar duration distribution, but also a similar structure (please look at Figure 4.3 in the main paper for a full comparison between D and M, and refer to Figure 4.23 to find an example

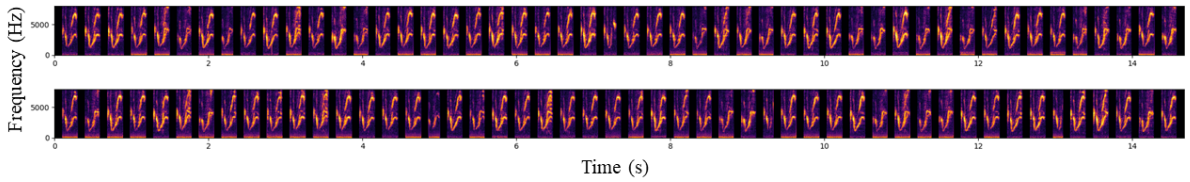


Figure 4.28: **Example of single syllables M.** 100 random selected syllables from the totality of the phrases belonging to class M.

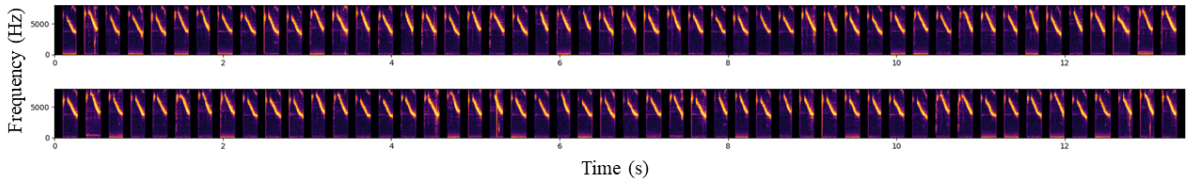


Figure 4.29: **Example of single syllables N.** 100 random selected syllables from the totality of the phrases belonging to class N.

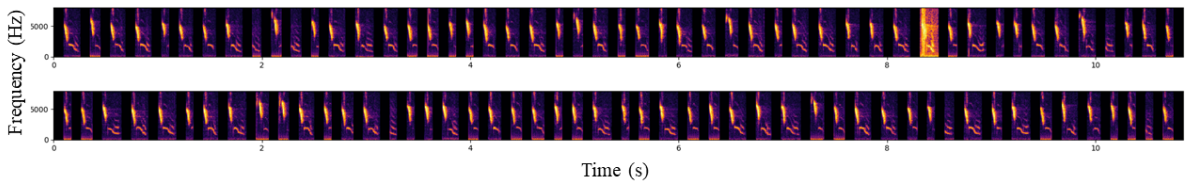


Figure 4.30: **Example of single syllables O.** 100 random selected syllables from the totality of the phrases belonging to class O.

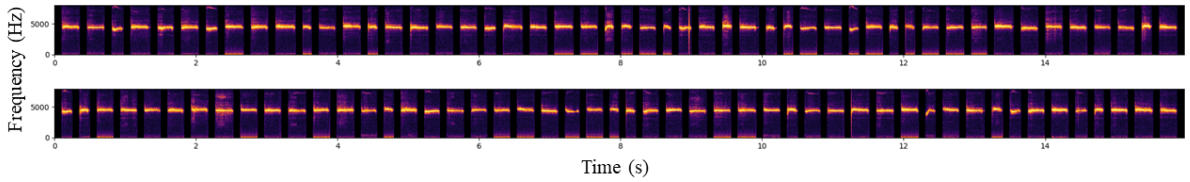


Figure 4.31: **Example of single syllables Q.** 100 random selected syllables from the totality of the phrases belonging to class Q.

of such error). To understand if this type of error can affect our work, we need to think about the use we want to make of the training dataset (e.g. which network we want to train and its characteristics). On the one side, we aim to use the dataset to train the generator of the GAN to produce realistic samples: from the structure of the network, we

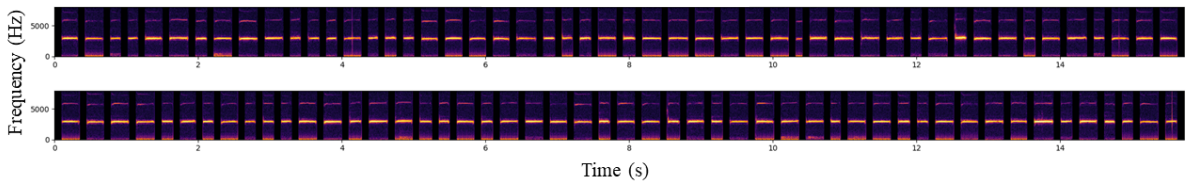


Figure 4.32: **Example of single syllables R.** 100 random selected syllables from the totality of the phrases belonging to class R.

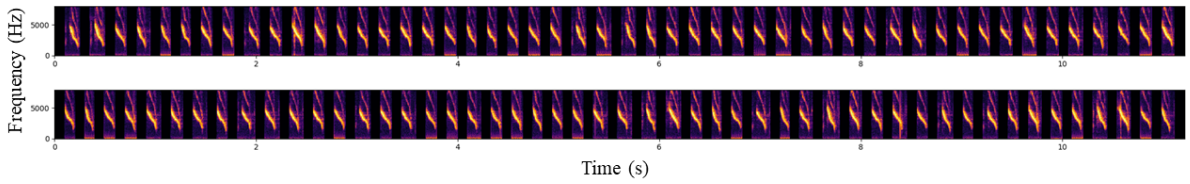


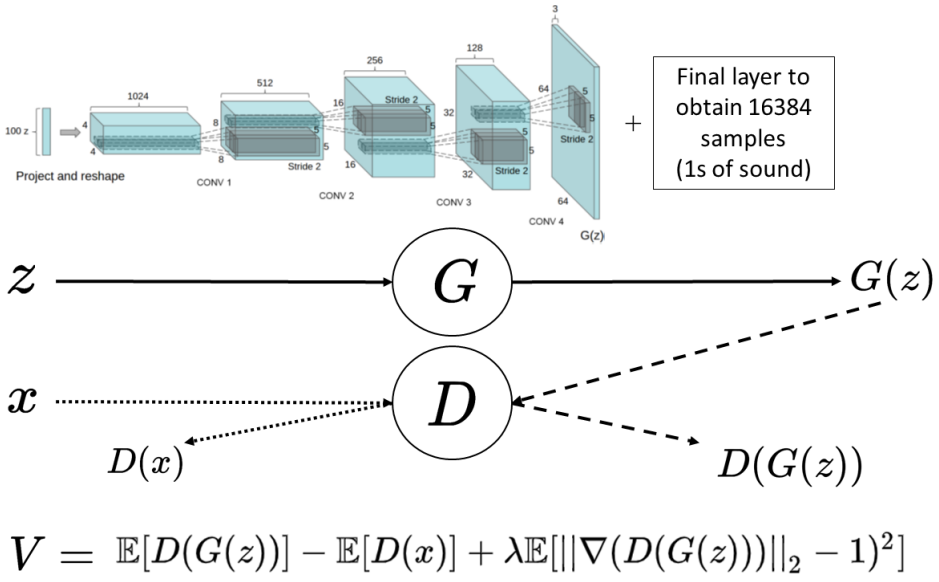
Figure 4.33: **Example of single syllables V.** 100 random selected syllables from the totality of the phrases belonging to class V.

know that the generator does not care about the class of each samples (indeed, it does not know the distribution of the training data). Since the generator does not have access to the classification, it is not a problem to have well selected samples in the wrong class. On the other side, we want to use the dataset to train a classifier able to determine for each sample the class it belongs to. We need to have a clean dataset to be able to teach the classifier. Nevertheless, after a visual inspection of the samples and after a validation of the classifier using the training dataset, we can assume that this type of error represent a low percentage of the total amount of error.

4.2 Appendix II: WaveGAN architecture

Figure 4.34 shows the structure of the generator G and the discriminator D, highlighting the architecture, the input and the output of the models, and the value function definition (i.e., V). The training data are pre-processed and stored in a tuple of *np.float32* tensors representing audio waveforms (x in Figure 4.34). The generator model G takes as input the latent vector $z \sim U(-1, 1)$. In the original paper, z is an 100-dimensional vector, but

we will use several lower dimensional inputs in this study. The upper part of Figure 4.34 shows the architecture of G: it has been taken from DCGAN (Radford et al., 2015) generator and it has been modified with an additional layer to obtain as output 16384 samples (i.e., 1 s of sound). Similarly, the discriminator model D takes as input vectors of length 16384 and gives as output an object of shape $(16, 1024)$. D can take as input the training data x or the output of the generator $G(z)$, respectively resulting, after deconvolution, in $D(x)$ and $D(G(z))$. These two quantities are the variables of the value function V. To define V, Donahue et al. (2018) removed the batch normalization from the generator and discriminator, and they used WGAN-GP (Gulrajani et al., 2017) strategy during training. This strategy consists in the introduction of a gradient penalty term in the loss function, driven by the regularization hyperparamaeter λ.



$$V = \mathbb{E}[D(G(z))] - \mathbb{E}[D(x)] + \lambda \mathbb{E}[||\nabla(D(G(z)))||_2 - 1)^2]$$

Figure 4.34: **WaveGAN architecture**. The architecture of the generator model G is the same architecture used in DCGAN (Radford et al., 2015) with an additional convolutional layer to obtain 16384 samples (i.e., 1 s of sound). The generator takes a latent vector $z \sim U(-1, 1)$ as input. The discriminator model D takes alternatively the training data x and the output of the generator $G(z)$ as input. After deconvolution, a representation of shape $(16, 1024)$ is obtained. The outputs of the discriminator, $D(x)$ and $D(G(z))$ are used as variable of the value function V (Gulrajani et al., 2017). Image adapted from Radford et al. (2015)

.

4.3 Appendix III: Classifier

4.3.1 Preliminary classifier

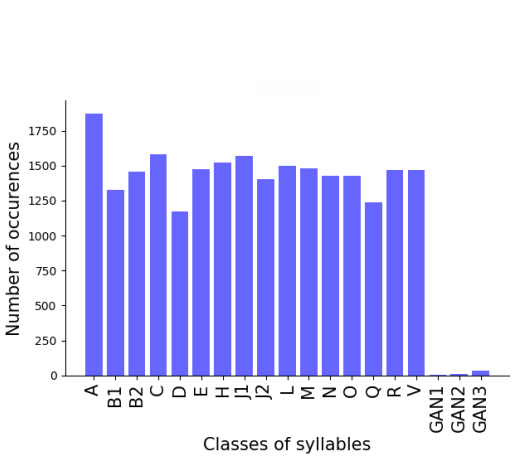
We first trained a classifier using the preliminary dataset described in Appendix 4.1. The classifier is able to differentiate 19 classes: the 16 classes of the repertoire and three alternative unknown classes: *GAN1*, *GAN2* and *GAN3*. We refer to this model as to *classifier-PRE*, where *PRE* stands for preliminary (since we used the preliminary dataset described in Appendix 4.1 to train it). As explained in the main paper, to train the classifier to recognize the unknown classes, in addition to the usual training dataset, we used three additional sets of generated samples. In summary, the unknown classes and the corresponding dataset we used to train the classifier are listed below:

- *GAN1*: $1k$ samples of early generations, obtained after ~ 3 epochs of one instance of a 3-dimensional GAN;
- *GAN2*: $1k$ samples of early generations, obtained after ~ 100 epochs of one instance of a 1-dimensional GAN;
- *GAN3*: $1k$ samples of early generations, obtained after ~ 100 epochs of one instance of a 3-dimensional GAN;

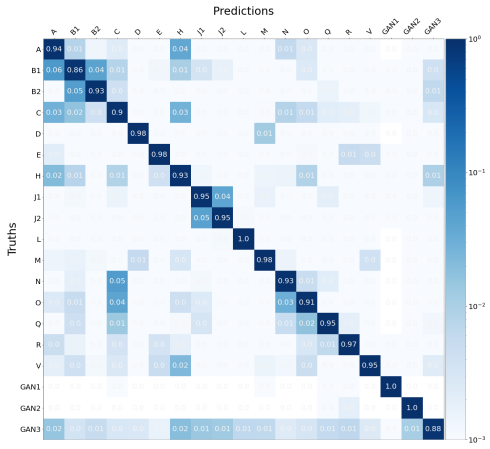
The model *classifier-PRE* is able to well recognise the 16 classes of the repertoire, but several classification errors can be observed (i.e., the 16 syllables are not balanced) in the left bottom panel (Figure 4.35). This could be a result of using a not cleaned dataset for training. In any case, as for *classifier-EXT*, the alternative unknown classes (*GAN1*, *GAN2* and *GAN3*) are only represented by a minority of samples.

4.3.2 Robustness of the classifier

The evaluation of the robustness of the three classifier (*classifier-PRE*, *classifier-REAL* and *classifier-EXT*) has been performed using a 10 folds cross-validation over the three corresponding training datasets. For each fold, 5 different instances of each classifier were



(a) Distribution of the preliminary training data obtained from *classifier-PRE*.



(b) Confusion matrix *classifier-PRE.*.

Figure 4.35: **Performance of the classifier on the training data**. The upper left panel shows how *classifier-REAL* performs on the training dataset. The average number of syllables per class is $\sim 1k$ for each of the 16 classes of the repertoire. This is coherent with the fact that we are considering a balanced dataset containing $\sim 1k$ sample per class. The lower left panel shows how *classifier-PRE* performs on the training dataset. As for *classifier-EXT*, the alternative unknown classes (*GAN1*, *GAN2* and *GAN3*) introduce a diffulty for the classifier. Although *classifier-PRE* suffers from a not cleaned training dataset (resulting in classification errors), it is able to well recognise the 16 classes of the repertoire. The upper right panel shows the confusion matrix (i.e. the level of confidence of the classifier in making the correct assignment) relative to *classifier-REAL*.

randomly initialized, trained and tested. The models were evaluated using an accuracy measure.

The accuracy has been defined as:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{timesteps}} \sum_{i=0}^{n_{timesteps}} \mathbb{1}(y(i) - \hat{y}(i)) \tag{4.5}$$

where $\mathbb{1}(x)$ is the indicator function, and considering a sequence of data x of length $n_{timesteps}$, a sequence of labels y and a sequence of classifier outputs $\hat{y}$, both also of length $n_{timesteps}$. The accuracy represents the capability of the classifier of making correct assignments for each time steps of MFCCs encoding the audio signal. An accuracy score close to 1 therefore indicates that the classifier is able to recognize the syllable category

of each sample and to correctly assign this category to each time steps representing this sample.

Both *classifier-REAL* and *classifier-REAL* show an higher level of accuracy with respect to *classifier-PRE* (Figure 4.36). As summarized in Table 4.3 the mean accuracy for the validation set is 0.9460 ± 0.0032 for *classifier-PRE*, 0.9815 ± 0.0024 for *classifier-REAL* and 0.9756 ± 0.0025 for *classifier-EXT*. The lower accuracy found for *classifier-PRE* could be related to the fact that it was trained on the preliminary training dataset described in Appendix 4.1 which contains a high number of errors in the ground truth.

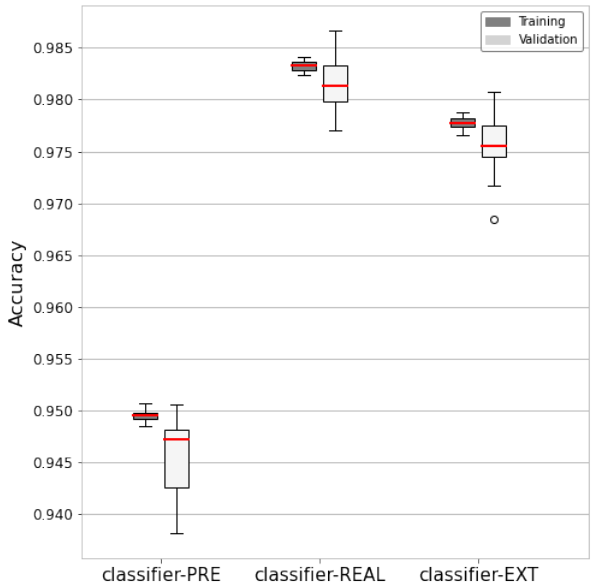


Figure 4.36: **Accuracy of the classifiers.** Comparison between *classifier-PRE* (on the left), *classifier-REAL* (in the middle) and *classifier-EXT* (on the right) in terms of the accuracy. For each model, the accuracy has been computed for the training set (gray rectangles) and for the validation set (white rectangles). The red lines represent the median accuracy relative to each set for each model. The white point visible for *classifier-EXT* represents an outlier, determined by the fact that the accuracy distribution is sharp. The highest accuracy has been reached with *classifier-REAL* where no alternative unknown classes have been introduced. The lowest accuracy has been reached with *classifier-PRE* which has been trained with a preliminary non-cleaned dataset (see Appendix 4.1).

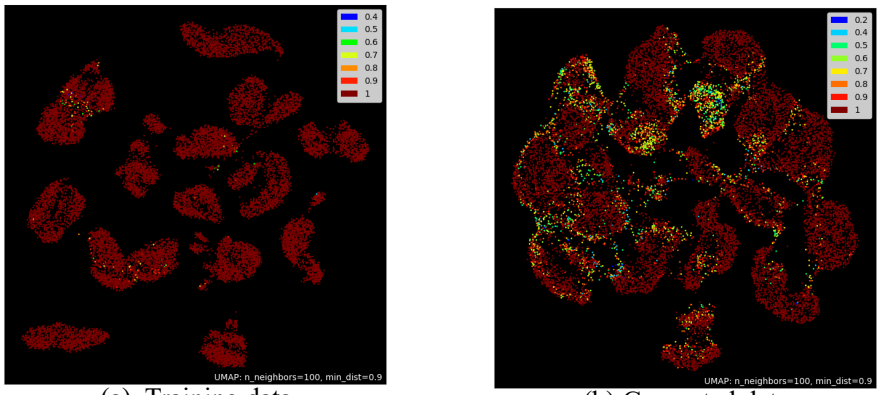
	Mean accuracy	
Classifier	Train	Validation
classifier-PRE	0.9495 ± 0.0005	0.9460 ± 0.0032
classifier-REAL	0.9832 ± 0.0005	0.9815 ± 0.0024
classifier-EXT	0.9777 ± 0.0006	0.9756 ± 0.0025

Table 4.3: **Mean accuracy.** Comparison between *classifier-PRE* (on the left), *classifier-REAL* (in the middle) and *classifier-EXT* (on the right) in terms of the median of the accuracy. For each model, the accuracy has been computed for the training set and for the validation set. The highest mean value has been reached with *classifier-REAL* where no alternative unknown classes have been introduced. The lowest mean value has been reached with *classifier-PRE* which has been trained with a preliminary non-cleaned dataset (see Appendix 4.1).

4.3.3 Certainty of the classifier

For each syllable, the classifier described in Section 4.3.4 provides a distribution which determines to which class the classifier assign that particular syllable. As mentioned, a soft-max operation is then applied in order to obtain a distribution bounded between 0 and 1. That is, each syllable is assigned to a particular class with a probability given by the max value of the resulting N-dimensional vector, where N is the number of classes present in the vocabulary. For instance, $N = 21$ for *classifier-EXT*. Such a classifier gives high scores for the training data, for which the majority of the syllables is assigned to a class with $p_s > 0.9$ (brown points in Figure 4.37a). Only a few syllables are assigned to a class with a probability $p_s \leq 0.9$.

Differently, *classifier-EXT* shows a higher uncertainty while classifying the generated data (Figure 4.37b). Although the majority of the syllables is assigned to a class with $p_s > 0.9$ (brown points), the number of syllables assigned to a particular class with a lower probability increases.



(a) Training data (b) Generated data

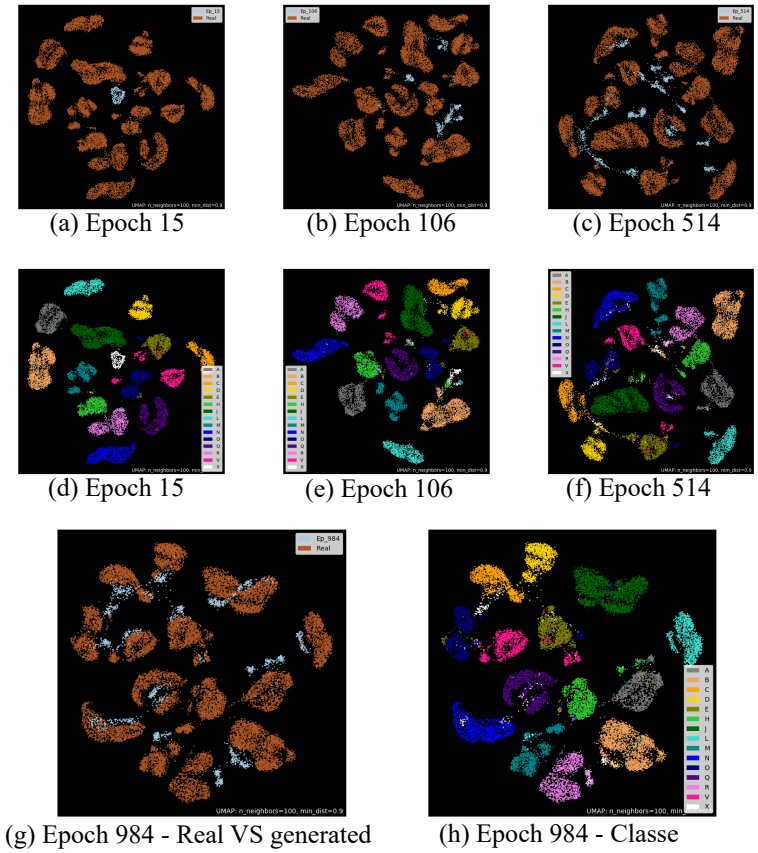
Figure 4.37: **Certainty of the classifier**. Probability of each syllable (each point) to belong to a particular class for the trainin data (a) and the generated data (b). Each color corresponds to an interval starting at the value indicated in the legend and ending at the next color value. For example, brown points belong to a certain class of the vocabulary with a probability $0.9 < p_s \leq 1$. An higher uncertainty of *classifier-EXT* can be observed when it is applied to the generated data: indeed, an higher number of syllables are assigned to a certain class with a probability $p_s \leq 0.9$ (all the points but the brown points).

4.4 Appendix IV: Extension of the qualitative analysis

4.4.1 Balanced UMAP representation

A similar UMAP representation to the one shown in Figure 4.12 can be obtained considering a balanced subset of $\sim$ 2100 generated syllables (100 per class) instead of $1k$ random generations at epochs 15, 106, 514 and 984. The fact that the generated syllables seem to be located at a bigger distance from the training data (Figure 4.38(a-c, g)) when considering a balanced dataset of 2100 samples (with respect to the same representation obtained from $1k$ random generated syllables shown in Figure 4.12(a-c, g)) can be related to the way UMAP determines the clusters. Indeed, as much generated data are given to UMAP as much the representation obtained takes into account the difference between the

real and the generated data, showing a less compact representation.



(a) Epoch 15 (b) Epoch 106 (c) Epoch 514

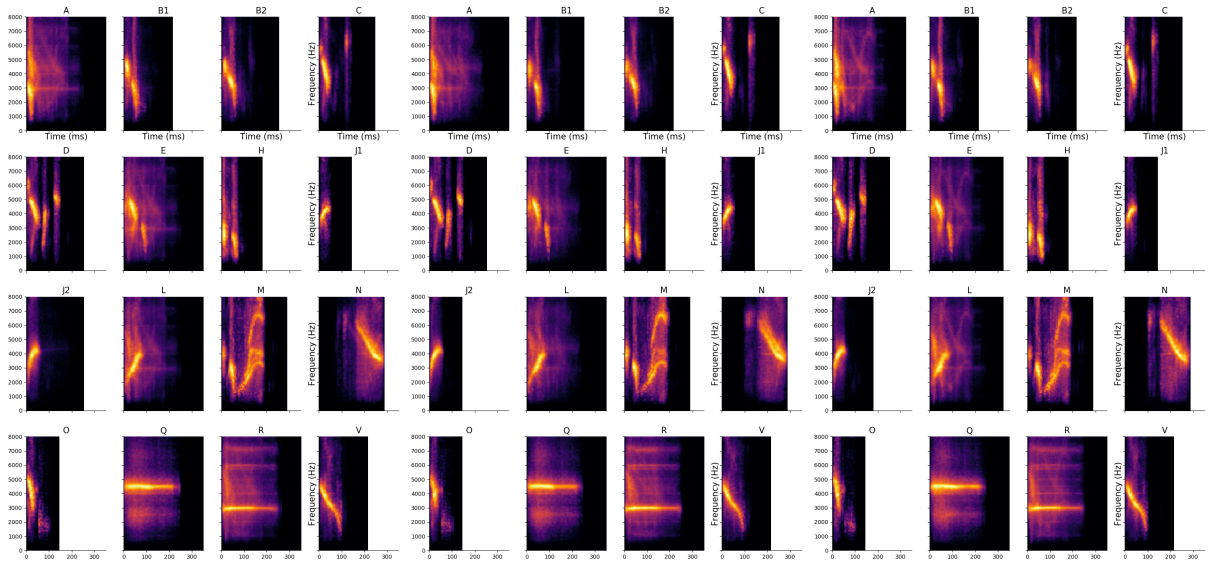
(d) Epoch 15 (e) Epoch 106 (f) Epoch 514

(g) Epoch 984 - Real VS generated (h) Epoch 984 - Classe

Figure 4.38: **Syllable space representation across time.** Syllable space representation obtained from the training dataset ($16k$ syllables) and 2100 syllables (100 per class, when the class is present, for a total of 21 classes - the 16 classes of the repertoire and 5 alternative unknown classes, here grouped as class X) generated at epochs 15, 106, 514 and 984 using UMAP (McInnes et al., 2018). Panels (a-c) and (g) show the training data (brown points) and the generated data (blue points). These four figures are different because the analyzed dataset differs for the 2100 generations specific of each epoch. Panels (d-f) and (h) show the same representation of panels (a-c) and (g) with the classes of syllables visible. Each cluster/color corresponds to one class of the repertoire and class X (in white) represents the cumulative class of the alternative unknown classes (in this case, *EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*). Here, the training has been done using $ld = 3$, the generator of instance *Ex 6* has been used to generate the syllables and *classifier-EXT* has been used to identify both the generated data and the training data.

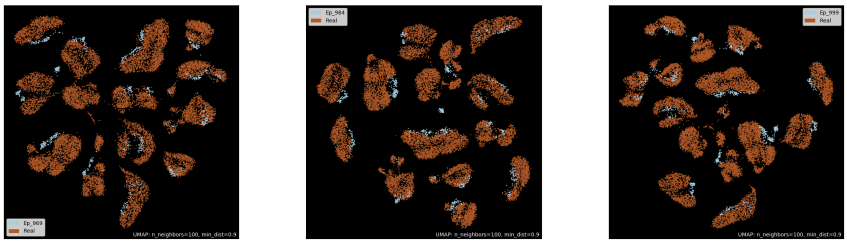
4.4.2 Stability of the training

The the mean spectrograms (Figure 4.39) and the UMAP representations (Figure 4.40) show a similar representation for consecutive epochs (i.e., epochs 969, 984 and 999) which show the stability of instance *Ex 6* around the convergence of the training. The differences in the UMAP representation (Figure 4.40) are given by the fact that the set of generated data is different at each epoch.

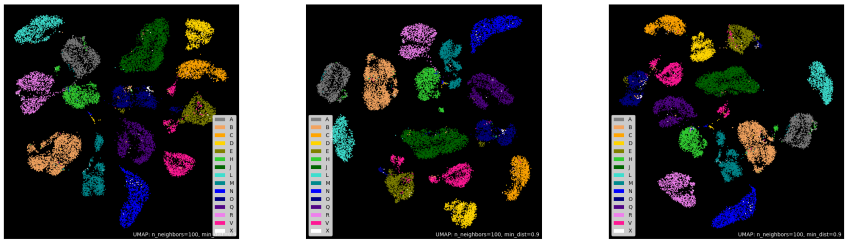


(a) Epoch 969 (b) Epoch 984. (c) Epoch 999.

Figure 4.39: **Stability of the mean spectrogram.** Mean spectrogram of $1k$ syllables generated at epochs 969 (a), 984 (b) and 999. The three epochs share a similar representation of the syllables. Here, the training has been done using $ld = 3$, the generator of instance *Ex 6* has been used to generate the syllables and *classifier-EXT* has been used to identify the generated data.



(a) Epoch 969 (b) Epoch 984 (c) Epoch 999



(d) Epoch 969 (e) Epoch 984 (f) Epoch 999

Figure 4.40: **Stability of UMAP representation.** UMAP representation of the training dataset ($16k$ syllables) and $1k$ generated data at epochs 969 (a-d), 984 (b-e) and 999 (d-f). The top panels (a-c) show the representation of the training data (brown points) and the generated data (light blue points) over time. The bottom panels (d-f) show the same representation highlighting the classes of the vocabulary. Each cluster/color correspond to one class of the repertoire and class X (in white) represents the cumulative class of alternative unknown classes (in this case, *EARLY15*, *EARLY30*, *EARLY45*, *OT* and *WN*). Although the representation is slightly different (given the fact that the generated syllables vary across time,it is possible to observe how the generated syllable mix well with the clusters obtained from the training data at all epochs), it remains stable across consecutive epochs. Here, the training has been done using $ld = 3$, the generator of instance *Ex 6* has been used to generate the syllables and *classifier-EXT* has been used to identify both the generated data and the training data.

4.4.3 Latent space exploration

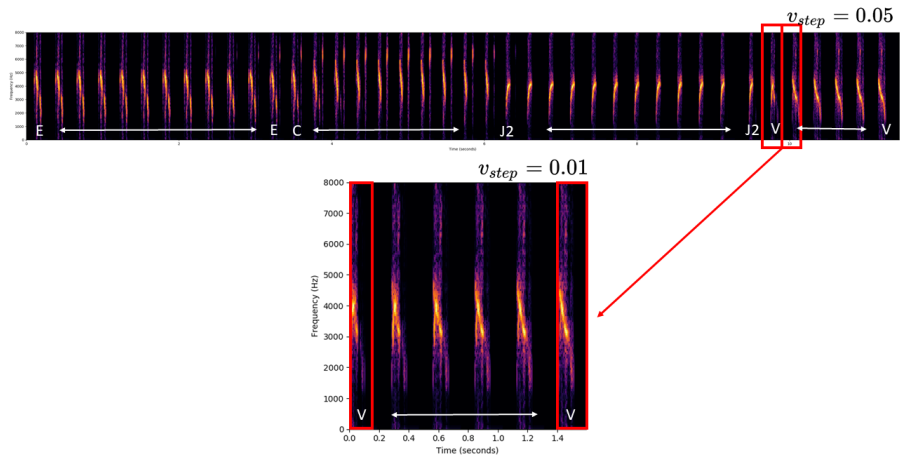


Figure 4.41: **Exploration of the latent space: one component variation. First component.**. We selected a random latent vector $z \sim R^3([-1, 1])$ to create a baseline syllable. Then, we moved one by one the components of the vector by a variation step equal to $v_{step} = 0.05$. We observed all the syllables produced to look at how they evolve and if there are non-smooth transitions. The upper panel of Figure 4.14 shows the exploration of the first component of the latent vector obtained with $v_{step} = 0.05$. The syllable V contained in the red square on the right represent a point where a non-smooth transition has been detected. For these particular transitions, we considered the two consecutive syllables obtained from the first variation step (i.e., two consecutive syllables contained in two consecutive red squares) and we applied a variation step of $v_{step} = 0.01$ to the first component of the latent vector to generate intermediate syllables. The bottom panel show the exploration of the non-smooth transition highlighted above. The syllables have been obtained the 3-dimensional generator obtained from instance *Ex 6* and the name of the syllable for this analysis has been provided by *classifier-EXT*.

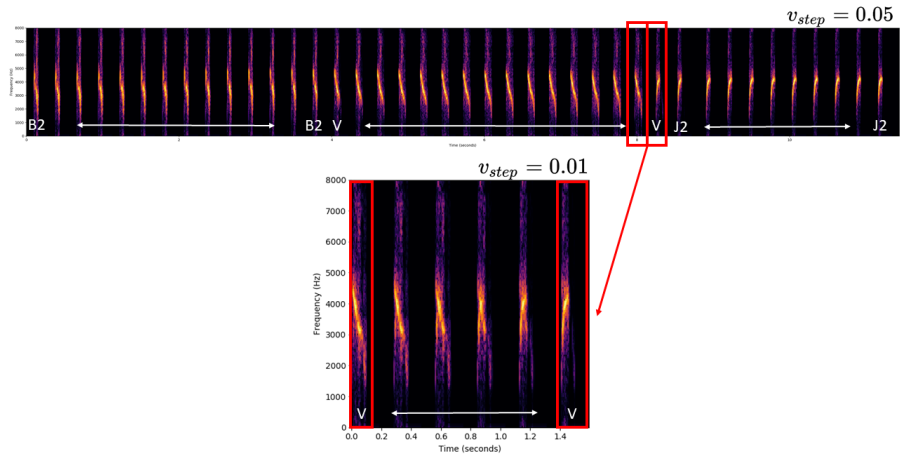


Figure 4.42: **Exploration of the latent space: one component variation. Second component.**. We selected a random latent vector $z \sim R^3([-1, 1])$ to create a baseline syllable. Then, we moved one by one the components of the vector by a variation step equal to $v_{step} = 0.05$. We observed all the syllables produced to look at how they evolve and if there are non-smooth transitions. The upper panel shows the exploration of the second component of the latent vector obtained with $v_{step} = 0.05$. The syllable V contained in the red square on the right represent a point where a non-smooth transition has been detected. For these particular transitions, we considered the two consecutive syllables obtained from the first variation step (i.e., two consecutive syllables contained in two consecutive red squares) and we applied a variation step of $v_{step} = 0.01$ to the first component of the latent vector to generate intermediate syllables. The bottom panel show the exploration of the non-smooth transition highlighted above. The syllables have been obtained the 3-dimensional generator obtained from instance *Ex 6* and the name of the syllable for this analysis has been provided by *classifier-EXT*.

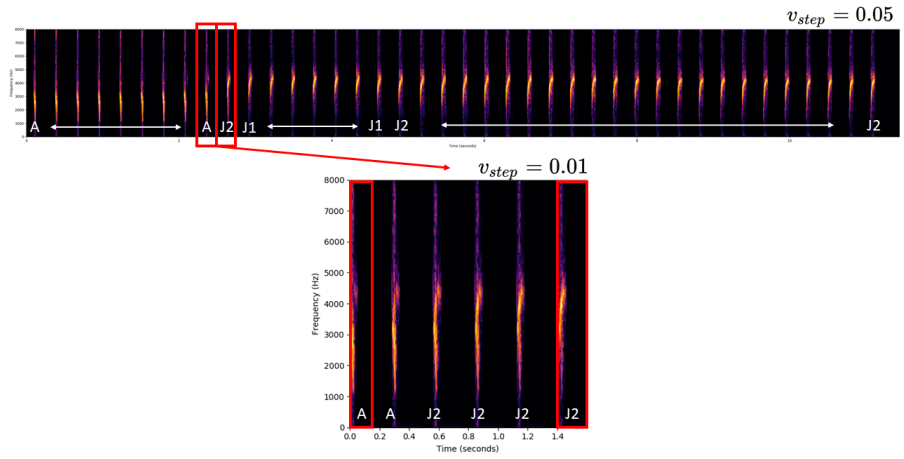


Figure 4.43: **Exploration of the latent space: one component variation. Third component.**. We selected a random latent vector $z \sim R^3([-1, 1])$ to create a baseline syllable. Then, we moved one by one the components of the vector by a variation step equal to $v_{step} = 0.05$. We observed all the syllables produced to look at how they evolve and if there are non-smooth transitions. The upper panel shows the exploration of the third component of the latent vector obtained with $v_{step} = 0.05$. The syllable V contained in the red square on the right represent a point where a non-smooth transition has been detected. For these particular transitions, we considered the two consecutive syllables obtained from the first variation step (i.e., two consecutive syllables contained in two consecutive red squares) and we applied a variation step of $v_{step} = 0.01$ to the first component of the latent vector to generate intermediate syllables. The bottom panel show the exploration of the non-smooth transition highlighted above. The syllables have been obtained the 3-dimensional generator obtained from instance *Ex 6* and the name of the syllable for this analysis has been provided by *classifier-EXT*.

4.4.4 Preliminary analysis

In a preliminary analysis, we used the preliminary training dataset described in Appendix 4.1 to train WaveGAN. Then, we used the experimental setup described in Section 4.3.2. We used the preliminary classifier described in Appendix 4.3.1 to identify the syllables as elements of 19 classes: the 16 classes of the repertoire and 3 alternative unknown classes $x \in X$ (*GAN1*, *GAN2* and *GAN3*). Further details about these classes can be found in Appendix 4.3.1.

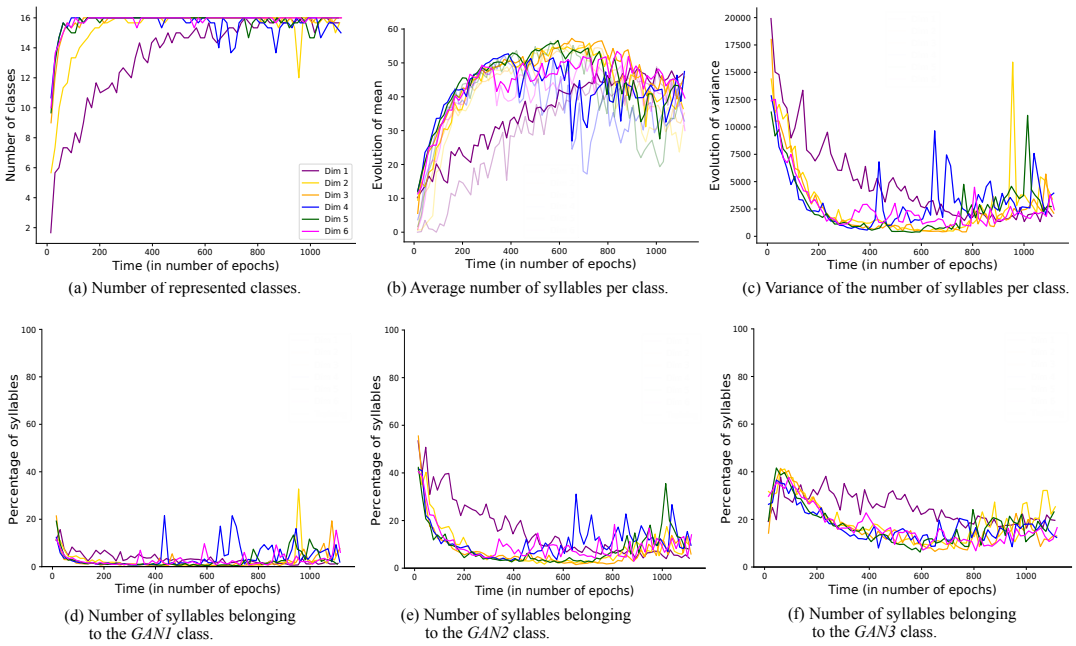
Latent space dimension

Complete version of Figures 4.17 (Figure 4.44). Accordingly to what observed in the main paper for Figure 4.44(a-c) and similarly to what shown in Figure 4.8, the percentage of alternative unknown syllables (here, *GAN1*, *GAN2* and *GAN3*) decreases over time, and eventually increases when overtraining begins (e.g., the green line in Figure 4.44e).

The fact that the performance obtained for $ld = 3$ is comparable with the performance obtained for $3 > l2 \leq 6$ and better than the performance obtained for $1 \leq ld \leq 2$ (Figure 4.44) is also confirmed by a better mean spectrogram representation 4.45. A good epoch of training is determined by looking at the classifier distribution (shown on the top of each spectrogram in Figure 4.45. The mean spectrograms obtained for $ld = 1$ (Figure 4.45a) and $ld = 2$ (Figure 4.45b) show noisy representations for several syllables. For instance, but not restricted to, syllables *C*, *H*, *O*. Nevertheless, some syllable representation are influenced by the fact that the trainings used to generated the samples have been done using the preliminary dataset described in Appendix 4.1.

Dataset size

Complete version of Figure 4.18 (Figure 4.46). Accordingly to what observed in the main paper for Figure 4.44(a-c) and similarly to what shown in Figure 4.8 and Figure 4.44, the percentage of alternative unknown syllables (here, *GAN1*, *GAN2* and *GAN3*) decreases over time, and eventually increases when overtraining begins (e.g., the blue line aroung epoch 900 in Figure 4.44e).



(a) Number of represented classes.
(b) Average number of syllables per class.
(c) Variance of the number of syllables per class.
(d) Number of syllables belonging to the *GAN1* class.
(e) Number of syllables belonging to the *GAN2* class.
(f) Number of syllables belonging to the *GAN3* class.

Figure 4.44: **Comparison between different latent space dimension: quantitative measure.**.Each line represents the average over 3 instances of training at a particular latent space dimension. Panel (a) shows how many syllables of the repertoire are covered by the generator across time. Panel (b) shows how many syllable per class have been generated in average. The dark lines show the evolution of the mean, whereas the light lines shows the evolution of the median. To build these two panels (a) and (b), only the repertoire's classes have been taken into account. Panel (c) shows the evolution of the variance of how many syllable per class have been produced. Panels (d-f) show the percentage of generated syllables belonging to classes *GAN1*, *GAN2* and *GAN3* across time in comparison with the percentage of syllables belonging to the same class in the training data. A 3-dimensional WaveGAN (orange line) reacheas convergence as good as higher-dimensional WaveGANs(blue, green and magenta lines) and better than lower dimensional WaveGANs (purple and yellow lines). We varied the latent space dimension as $ld = 1, 2, 3, 4, 5, 6$ and we kept fix the training dataset. The training of all the instances of WaveGAN has been done using the preliminary training dataset introduced in Appendix 4.1 and *classifier-PRE* (see Appendix 4.3.1) has been used to identify the syllables.

.

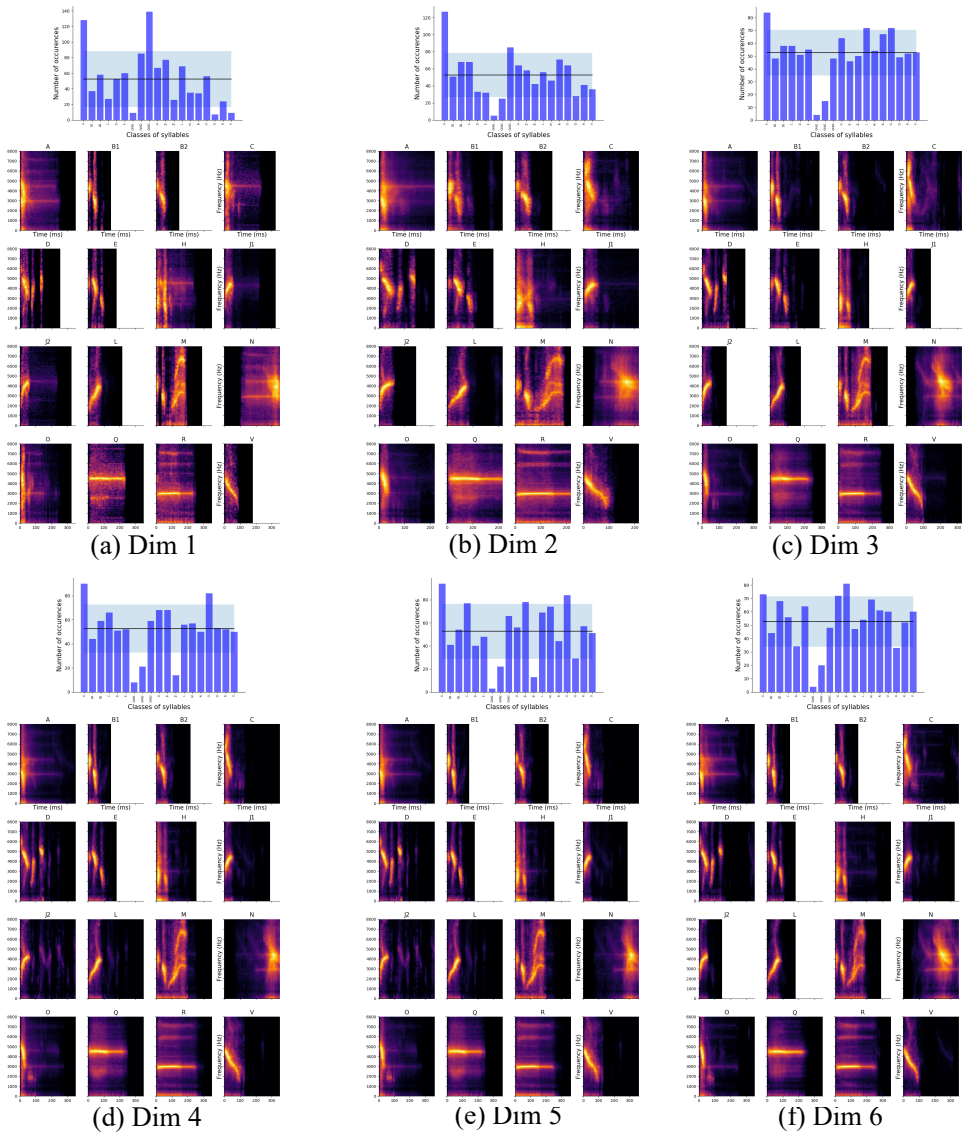
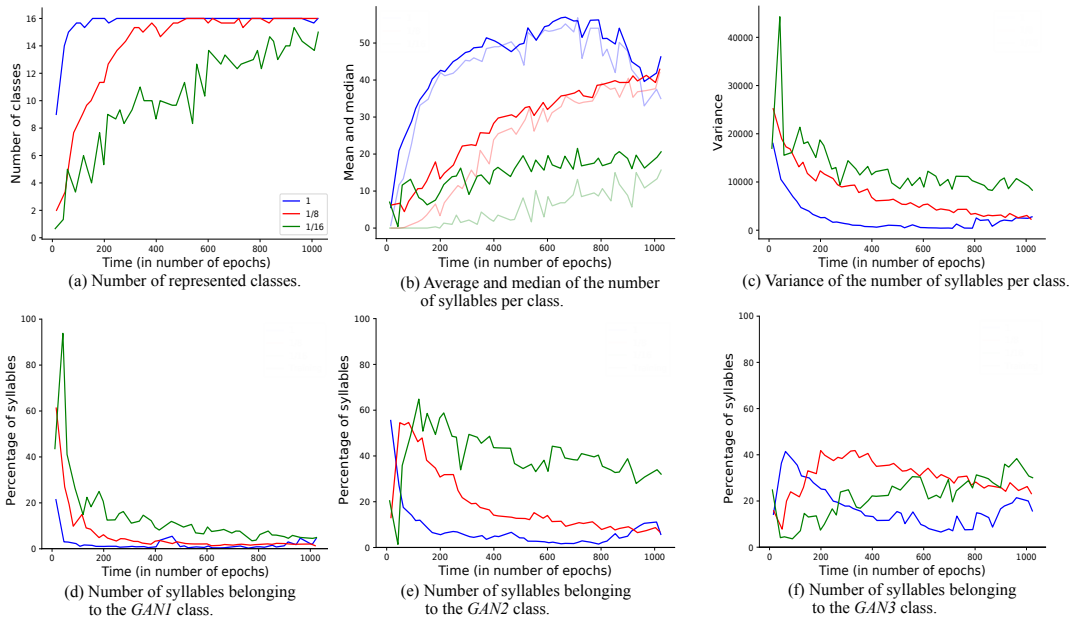


Figure 4.45: **Comparison between different latent space dimension: qualitative measure**. Mean spectrogram of $1k$ syllables generated at a good epoch of training determined by looking at the classifier distibution (on top of each spectrogram) for $ld = 1, 2, 3, 4, 5, 6$. Whereas the representations obtained for $ld = 1$ (a) and $ld = 2$ (b) show a noisy representation for several syllables (e.g., syllable M in (b)), $ld = 3$ (c) shows a representation comparable to the one obtained for higher conditions (d-f). We relate the noise shown by syllable N to the fact that these trainings have been done using the preliminary dataset described in Appendix 4.1. Here, *classifier-PRE* has been used to identify the generated data

.



(a) Number of represented classes. (b) Average and median of the number of syllables per class. (c) Variance of the number of syllables per class.

(d) Number of syllables belonging to the *GAN1* class. (e) Number of syllables belonging to the *GAN2* class. (f) Number of syllables belonging to the *GAN3* class.

Figure 4.46: **Comparison between datasets of a different size**. We varied the dataset size as $d = 23456, 3600, 1600$ and we kept fix the latent space dimension at $ld = 3$. Each line in this figure represents the average over 3 instances of training at a particular dataset size condition. Panel (a) shows how many syllables of the repertoire are covered by the generator across time. Panel (b) shows how many syllable per class have been generated in average. The dark lines show the evolution of the mean, whereas the light lines shows the evolution of the median. To build these two panels (a) and (b), only the repertoire's classes have been taken into account. Panel (c) shows the evolution of the variance of how many syllable per class have been produced. Panels (d-f) show the percentage of generated syllables belonging to each garbage class (*GAN1*, *GAN2* and *GAN3*) across time in comparison with the percentage of syllables belonging to each garbage class in the training data. A dataset of bigger size (blue line) allows better and faster convergence than having a dataset of lower sizes (red and green lines). The training of all the instances of WaveGAN has been done using the preliminary training dataset introduced in Appendix 4.1 and *classifier-PRE* (see Appendix 4.3.1) has been used to identify the syllables.

.

Chapter 5

Canary sensorimotor model with a low-dimensional GAN generator

Contents

5.1 Introduction

The basic structure of a vocal learning schema involves three spaces (motor, sensory, perceptual), the motor control function, the sensory response function, and the learning architecture (Pagliarini et al., 2020). The simpler model developed in this thesis (see Chapter 3) does not implement a motor control function (i.e., the sensory space coincides with the motor space and there is no sound production). More complex models define a motor control function that allows the production of sound. The sensory response function processes the sound and defines the perceptual space. Finally, the learning architecture defines the goal of the learning, the learning algorithm, and, eventually, the exploration strategy.

The objective of this chapter is to define a vocal learning model that mimics the sensorimotor phase of learning in songbirds. The model should contain the three spaces mentioned above and must be able to produce a realistic sound as output. We aim to support the definition of this model with biological assumptions. At the same time, we aim to build a model able to learn in a finite amount of time.

In Chapter 3, the connections between the motor space and the perceptual space were learned using a theoretical inverse model (Pagliarini et al., 2018a) in a simple model where no distinction between the motor and the sensory space was made. That is, there was no need for defining a motor control function since there was no sound production. The simple model helped the understanding of how to introduce biological assumptions in a computational framework. On the one hand, even with a theoretical model, convergence problems arise from non-linearity. On the other side, it allowed the exploration of the structure of the network and, in particular, the dimension of the motor space: a high dimensional motor space leads to convergence in an unreasonable computational time (Pagliarini et al., 2018a).

In songbird literature, the motor control function has been often defined using a system of ordinary differential equations that model the anatomy of the syrinx (i.e., the birds' vocal organ) (Amador et al., 2013), or the features of sound (Doya and Sejnowski,

1998). Such models can provide qualitatively good productions but might be not able to perfectly reproduce the perceptuo-motor connections (Pagliarini et al., 2020). Usually, mechanistic models only use a few motor parameters to induce most of the changes in the output. Thus, it is hard to understand the mapping between motor parameters and output. Moreover, they are slow to simulate.

This chapter proposes to use the generator of WaveGAN (Donahue et al., 2018) to define the motor control function. The advantages of using generative neural networks are to obtain, on the one side, an uniformly distributed low-dimensional motor space and, on the other side, the resemblance of the generated data with the real data. These models have been introduced to solve tasks such as image, music, and speech generation or classification and have been used to investigate visual pathways in the brain (Ponce et al., 2019).

In this chapter, we define a complete vocal learning model. The model includes a motor, a sensory and a perceptual space, the motor control function, and the sensory response function. A 3-dimensional generative model (i.e., the latent space has dimension 3) introduced in Chapter 4 models the motor control function, while the combination of a Recurrent Neural Network-based classifier (as the one trained in Section **??** in Chapter 4) and a normalization layer models the sensory response function. The first problem we faced is dealing with a redundant motor space, i.e. multiple motor configurations correspond to the same sensory production. The goal of the model is therefore defined as a perceptual goal. The connections between the motor space and the perceptual space are learned through an inverse model. The motor space is explored using a random uniform exploration and a simple Hebbian learning rule drives the learning.

Section 5.2 introduces the vocal learning model schema and its components. Section 5.3 shows preliminary results about the influence of the learning rate when different sensory response functions are implemented. These results have been obtained under the simple assumptions described above and aim to be expanded. Section 5.4 summarizes the advantages of the tools we used to define the model and discusses the possible perspectives.

5.2 Methods

5.2.1 Network description

As for the simple theoretical model introduced in Chapter 3, a two-layer network models the connections between the perceptual space and the motor space. The first layer (on the left in Figure 5.1) represents the perceptual space. The second layer (on the right in Figure 5.1) represents the motor space. At each time t, the perceptual units are defined as a n_A-dimensional vector A_t, where n_A represents the size of the perceptual layer. The target set is the set of sounds (here, the N-dimensional repertoire of canary syllables) that we aim the model to learn. The motor units are defined as a n_M-dimensional vector M_t, where n_M represents the number of motor parameters. The synaptic weights at time t describing the connections between the motor and the perceptual space are defined by matrix W_t. Given M_t as input, the motor control function G_t provides a real sound S_t as output (i.e., an element of the sensory space). The sensory space is the domain of the sensory response function, that is, at each time t, the sensory response is a function of the actual sound production S_t: $A_t = f(S_t)$.

5.2.2 Motor control

The motor control function is needed to perform both the motor exploration during learning and to generate the goal syllables across time (to check how the learning goes on). In the proposed model, the generator part of a 3-dimensional GAN (i.e., $n_M = 3$) is used as motor control function. We decided to use a 3-dimensional generator to avoid a high dimensional motor space which could result in slower convergence (Pagliarini et al., 2018a). The generative model is fully described in Chapter 4. WaveGAN has been previously trained on a dataset of canary syllables to obtain a generator model able to provide good outputs (i.e., resembling the training data). Significantly, WaveGAN (Donahue et al., 2018) and GANs, in general, are characterized by a redundant input space (properly called *latent space*): one sensory output corresponds to multiple latent space

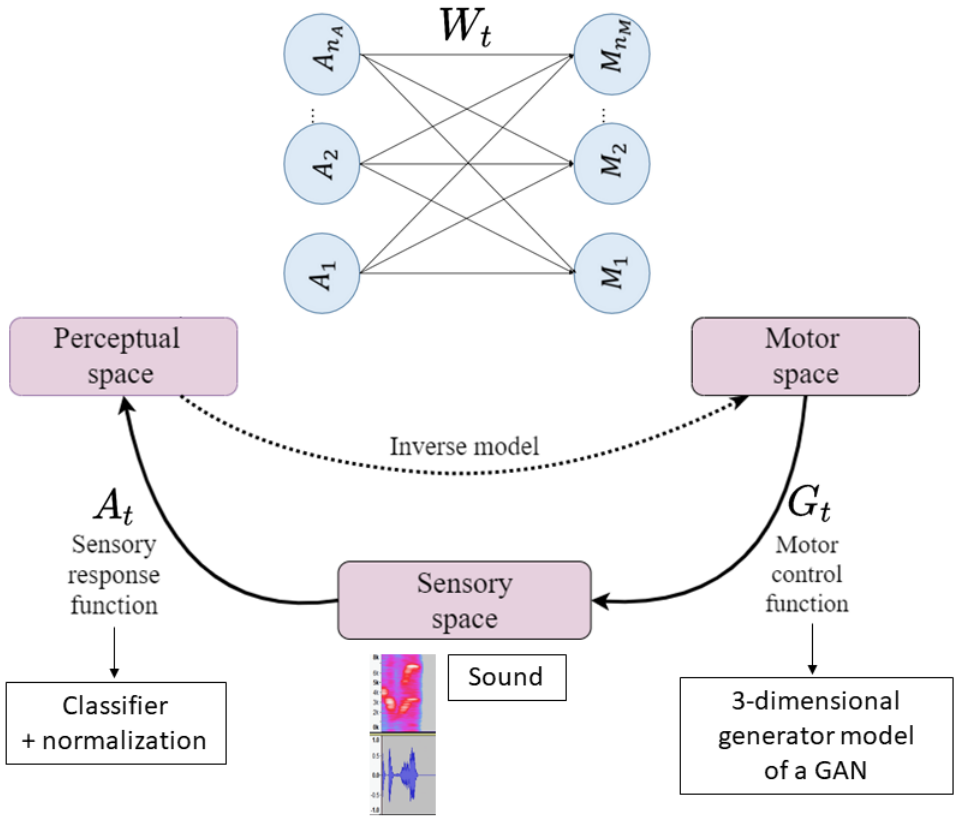


Figure 5.1: **Vocal learning model schema.**. The model contains three spaces: the perceptual space, the motor space, and the sensory space. A two-layer network connects the perceptual space (on the left) to the motor space (on the right). W_t represents the synaptic connections between perceptual and motor spaces at each time step t. The motor control function G is the generator part of the GAN trained in Chapter 4 that enables sound production. At each time step t, the sensory response A_t is a function of the actual sound production (i.e., $A_t = f(S_t)$, where S_t is the actual sound produced by G). The sensory response function is composed of a classifier (defined in Section 4.3.4 of Chapter 4), then a normalization layer to scale the obtained activation to values in $[0, 1]$.

vectors. In terms of the vocal learning model introduced in Section 5.2.1, one syllable type (i.e., belonging to one class of the repertoire) can be produced using multiple motor configurations.

5.2.3 Sensory system

The dimension of the perceptual space depends on how many different syllables the model learns at a time. The minimal dimension of the perceptual space is $n_A = 1$ when the model is learning only one syllable. The maximal dimension is $n_A = N$ when the model is learning all the syllables contained in the repertoire if it contains N different classes of syllables. The repertoire used to define the sensory target space is the same that has been used to train the generator model (i.e., the motor control function) and the classifier (i.e., the first layer of the sensory response function) in Chapter 4).

The sensory response function (i.e., A_t in Figure 5.1) is composed by two components. First, either *classifier-REAL* or *classifier-EXT* provides the sensory activation of a sound S_t (a syllable) over time for each class of the vocabulary. The vocabulary is classifier-specific. On the one hand, *classifier-REAL* returns as output a N-dimensional vector representing the likelihood that the syllable belongs to a given class for each of the N classes of the repertoire. On the other hand, *classifier-EXT* returns as output a $N + 1$-dimensional vector representing the syllable activation for each of the N classes of the repertoire and an alternative class X (i.e., a class defined by early and late generations of GAN generations, and artificial white noise). These classifiers have been defined in Section 4.3.4 of Chapter 4 and have been previously trained with the same dataset as the one used to train WaveGAN (to obtain the motor control function). Here, $N = 16$ and $X = 1$ (that is the cumulative class for multiple unknown classes). Further details about what X class represents are provided in Section 4.3.3 of Chapter 4. Then, a normalization layer rescales the obtained activation to values in $[0, 1]$: either (1) a softmax function provides a probability distribution representing the chance of the syllable to belong to each class of the vocabulary, or (2) a max-min scaling provides a normalized score (called *raw score*) which restricts the syllable activation to the range $[0, 1]$. The raw score is defined as

$$raw\_score = \frac{class_d - min(class_d)}{max(class_d) - min(class_d)}, \tag{5.1}$$

where $class_d$ represents the syllable activation vector provided by the classifier for one

syllable given as input. We will refer to (1) as *softmax score* and to (2) as *raw score*.

5.2.4 Learning algorithm

The model aims to learn $n_A = N$ different syllables. As mentioned in Section 5.2.2, the motor function G allows multiple motor configurations to produce the same sensory output. To deal with the redundancy of the motor control function, the goal is not represented by a motor target. Rather, it is a perceptual target. That is,

$$A^i_{t_f} \longrightarrow 1 \qquad for 1 \leq i \leq n_A = N, \tag{5.2}$$

where $A^i_{t_f}$ represents the sensory response relative to syllable i at time t_f (i.e., the last epoch of training), and N is the number of classes of the repertoire.

Learning is driven by the Hebbian learning rule

$$\Delta W_t \propto \eta M_t A_t, \tag{5.3}$$

where W_t represents the synaptic weight, η the learning rate, M_t the motor pattern at time t and A_t the sensory response at time t. The synaptic weights $W_{t=t_0}$ are initialized as random uniform values and vary according with Equation 5.3 until time $t = t_f$. We did not introduce a normalization in the learning rule: this choice is motivated by the fact that the motor space we consider in this model can not be normalized (otherwise, this would change the relationship between the motor command and the sensory output). The motor space is explored using random exploration.

5.2.5 Simulation details

We chose a set of $n_A = 16$ syllables as target set. This is the same repertoire as the one used in Chapter 4 to train the 3-dimensional GAN which serves as a motor control function and the classifier which serves as a sensory response function. At each time step t, a random motor vector M_t is given as input to the motor control function G. M_t is a 3-dimensional vector taking random values in $U[-1, 1]$. The motor control function G produces a syllable, which is given to the classifier as input and used to compute

the sensory response (i.e., the perceptual representation). For simplicity, we generated a priori a set of $5k$ motor vectors, the waveforms of the corresponding syllables, and the corresponding sensory response (both using the softmax and the raw score).

The synaptic weights $W \in M^{16\times3}$ are initialized as $W_{t_0} \in [-0.001, 0.001]$. We kept fixed the initial weights and the motor vector and we compared (1) the performance of the model when *classifier-REAL* and *classifier-EXT* are used as sensory response functions, (2) the performance of the model when different sensory responses (i.e., softmax score and *raw score*) are used to encode the probability of a syllable to belong to each class of the repertoire, and (3) the influence of using different learning rates ($\eta = 0.01$ versus $\eta = 0.1$). For each condition, we stop the learning at time $t = t_f = 1500$.

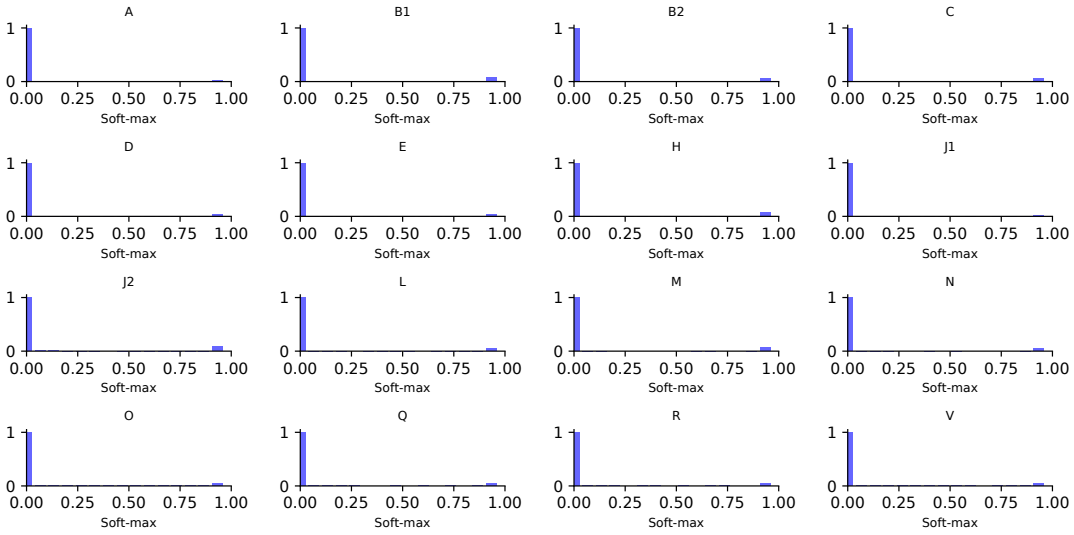
5.3 Results

5.3.1 Motor exploration

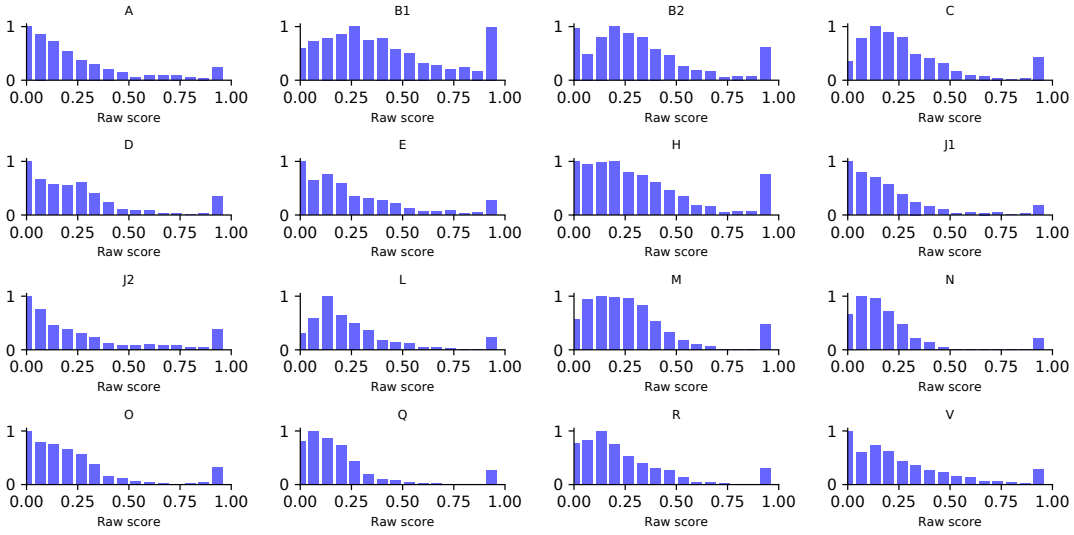
At each time t the model performs a random motor exploration M_t which enables the generations of a sound and, consequently, the sensory response to this sound is computed through the sensory function. Depending on how we compute the sensory response (either using the softmax score or the raw score defined by Equation 5.1), we obtain different activations in the perceptual layer. Please note that the architecture of the motor control function allows a sensory activation bigger than zero also at the beginning of the learning (when the synaptic connections are weak and close to 0). This is due to the fact that there is not a "neutral position" in the latent space: at any point of the space, the generator can produce a syllable that will activate (at least slightly) one class of the classifier.

When softmax score is used to model the sensory response, the majority of motor explorations performed (from the beginning of the learning until time $t_f = 1500$) show a very low sensory response (most of the time). The distribution of the sensory response does not show intermediate values between 0 and 1 both when *classifier-REAL* (Figure 5.2a) and *classifier-EXT* (Figure 5.3a) are used as sensory response function.

On the contrary, if the sensory function is computed as a normalized raw score (defined by Equation 5.1, the distribution of the sensory responses (relative to the motor explorations performed during learning) becomes more homogeneous. The usage of the raw score instead of the softmax score, allows a smoother perceptual encoding both when *classifier-REAL* (Figure 5.2b) and *classifier-EXT* (Figure 5.3b) are used as sensory response function. Figure 5.4 summarizes how many motor explorations obtained a sensory response in the range $[0.9, 1]$ over 1500 motor explorations performed during learning.

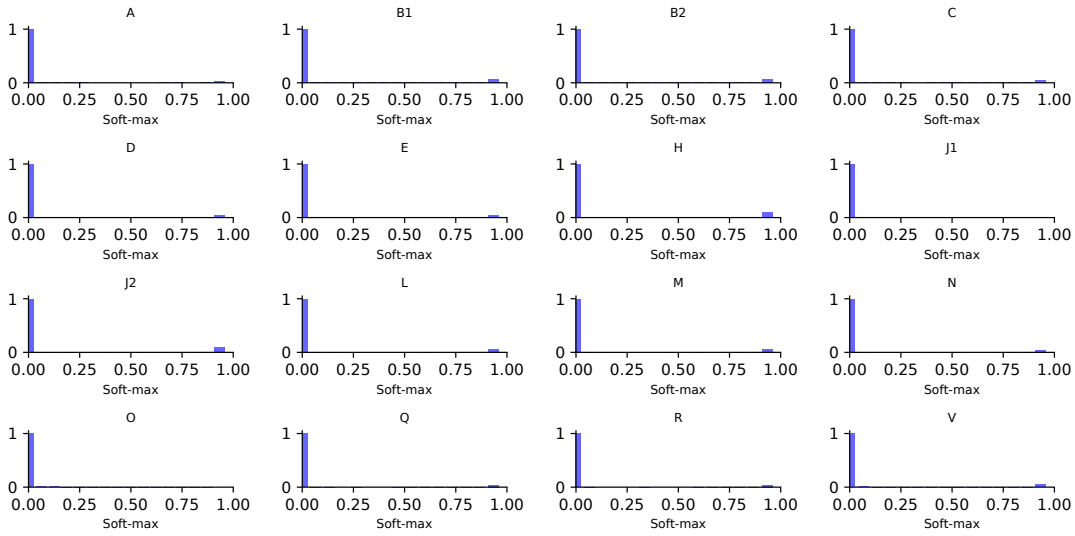


(a) Soft-max distribution obtained with *classifier-REAL* .

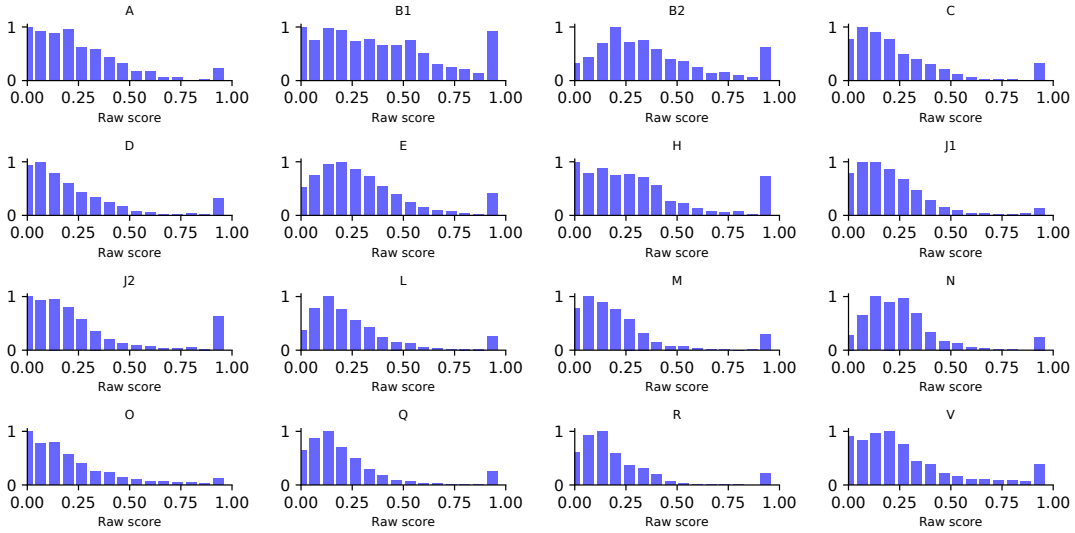


(b) Raw score distibution obtained with *classifier-REAL*.

Figure 5.2: **Motor exploration: *classifier-REAL***. Distribution of the sensory responses obtained at each time t from the random motor exploration when the normalization of the activation is obtained as a softmax score (a) and a raw score (b). The distribution of the sensory repsonse obtained using the softmax score concentrates around 0 and, eventually, shows values close to 1 (over 1500 motor explorations, $\sim$ 1000 are close to 0). The distribution of the sensory response obtained using the raw score (defined by Equation 5.1) is more homegeneous, especially for small values (smaller than 0.5). Here, $n_A = 16$, $n_M = 3$, $W_{t_0} \in [-0.001, 0.001]$, $t_f = 1500$, $\eta = 0.01$ and *classifier-REAL* is used as the first layer of the sensory response function.



(a) Soft-max distribution obtained with *classifier-EXT* .



(b) Raw score distibution obtained with *classifier-EXT*.

Figure 5.3: **Motor exploration: *classifier-EXT***. Distribution of the sensory responses obtained at each time t from the random motor exploration when the normalization of the activation is obtained as a softmax score (a) and a raw score (b). The distribution of the sensory repsonse obtained using the softmax score concentrates around 0 and, eventually, shows values close to 1 (over 1500 motor explorations, $\sim$ 1000 are close to 0). The distribution of the sensory response obtained using the raw score (defined by Equation 5.1) is more homegeneous, especially for small values (smaller than 0.5). Here, $n_A = 16$, $n_M = 3$, $W_{t_0} \in [-0.001, 0.001]$, $t_f = 1500$, $\eta = 0.01$ and *classifier-EXT* is used as the first layer of the sensory response function.

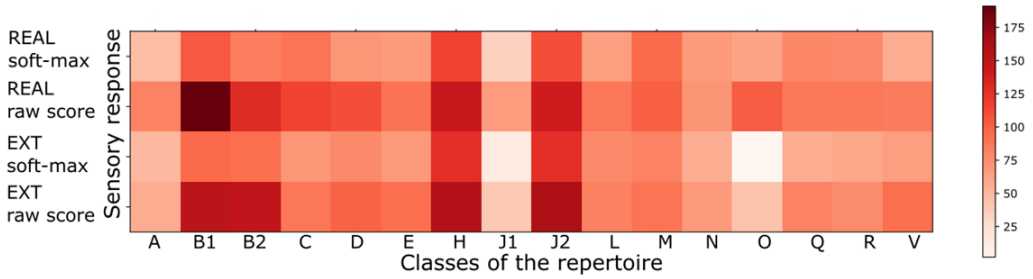
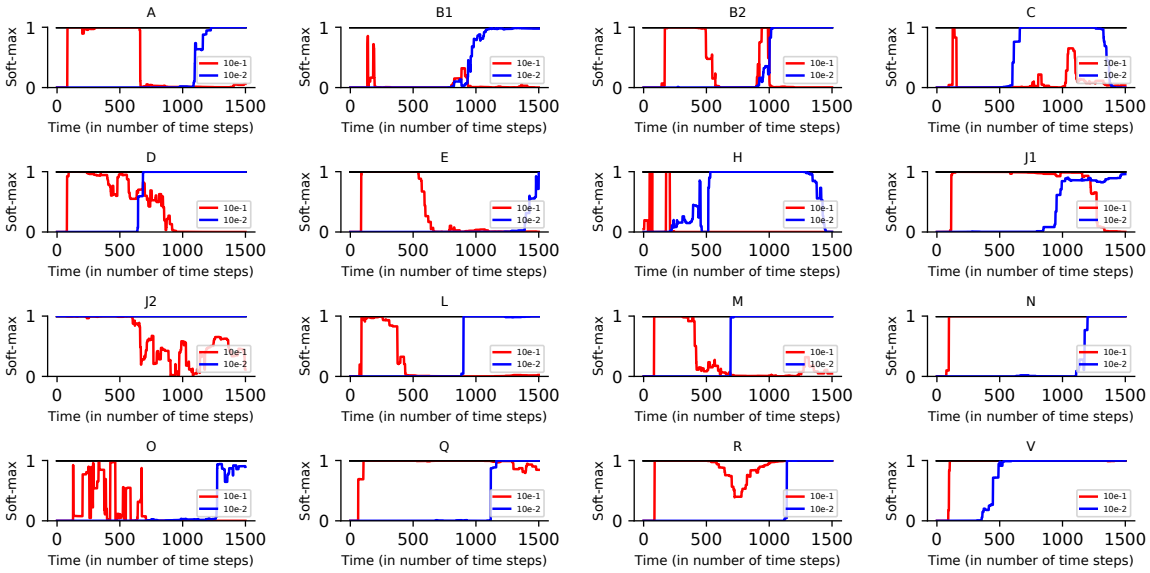


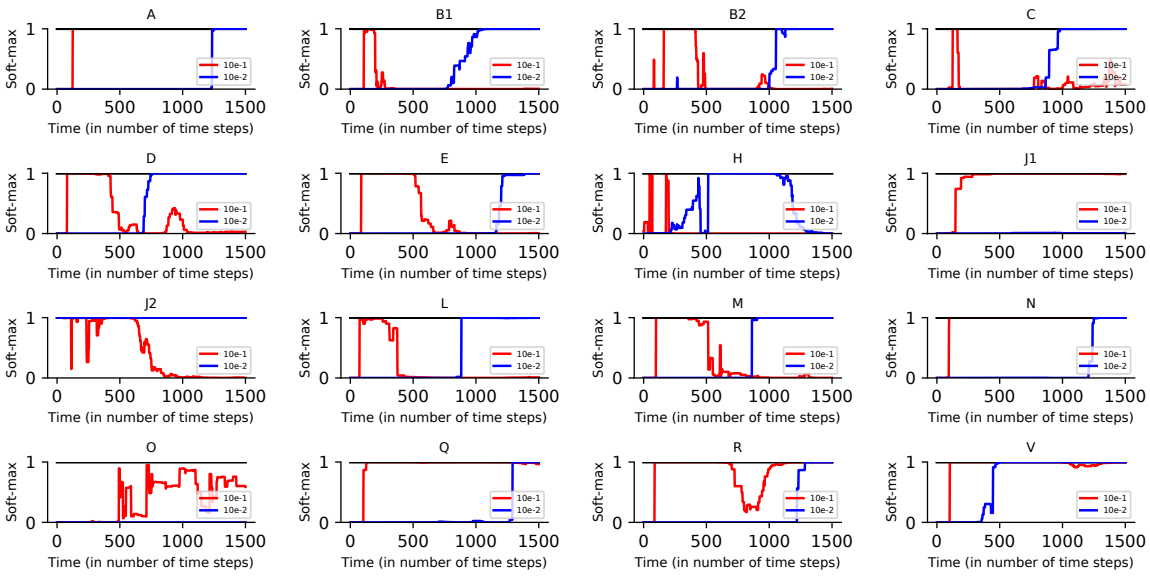
Figure 5.4: **Sensory response relative to the motor exploration.**. Total amount (out of 1500) of motor exploration having either a softmax score (first and third lines) or a raw score (second and fourth lines) higher than 0.9 per class of syllables in the repertoire. The sensory activation vector has been obtained either using *classifier-REAL* (first and second lines) or *classifier-EXT* (third and fourth lines).

5.3.2 Influence of the learning rate

A learning rate of $\eta = 0.1$ (red lines in Figures 5.5 and 5.6) can induce faster changes in the synaptic weights (i. e., in W_t) with respect to $\eta = 0.01$ (blue lines in Figures 5.5 and 5.6). One can see that a bigger learning rate induces a faster learning and thus a sharper increase (and an eventual decrease) of the perceptual activation of a given syllable. This is independent on which sensory response function we use (*classifier-REAL* or *classifier-EXT*) or which normalization we implement (the softmax score or the raw score). Instead, different ways to compute the sensory response normalization (softmax score versus raw score) result in different learning curves. A softmax score results in a highly non-linear learning curve: the sensory response stays low for several time steps after the beginning of the learning, and increases sharply after a number of time steps which differs from syllable to syllable (Figure 5.5). Afterward, the sensory response remains high and stable for a limited time before a new decay begins after a certain time t_{critic}. A faster decay is observed for $\eta = 0.1$ (red lines in Figure 5.5).

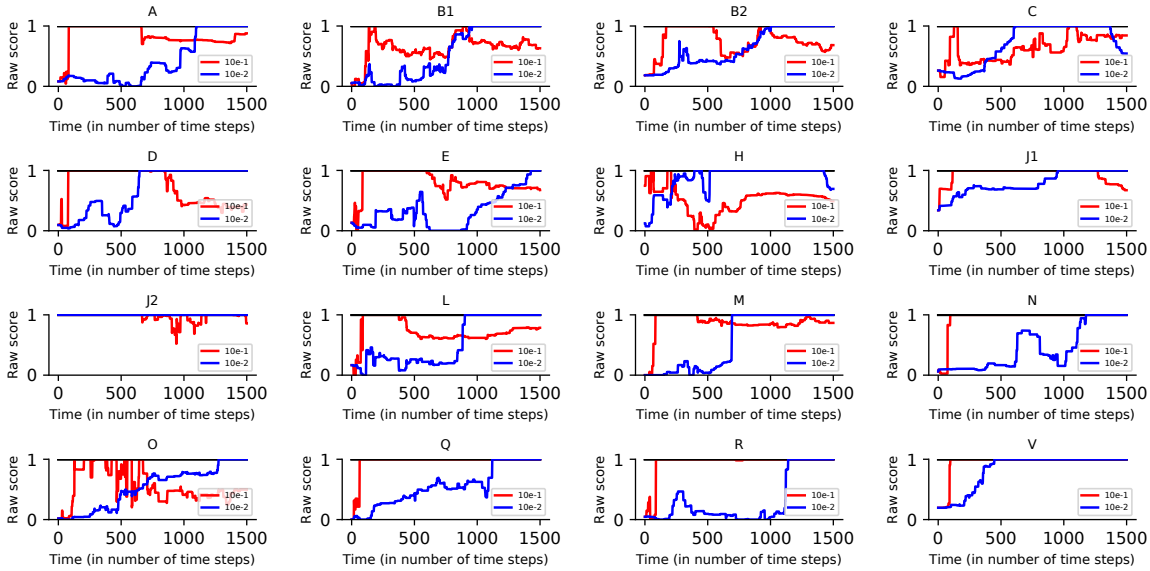


(a) Sensory response over time obtained from *classifier-REAL*.

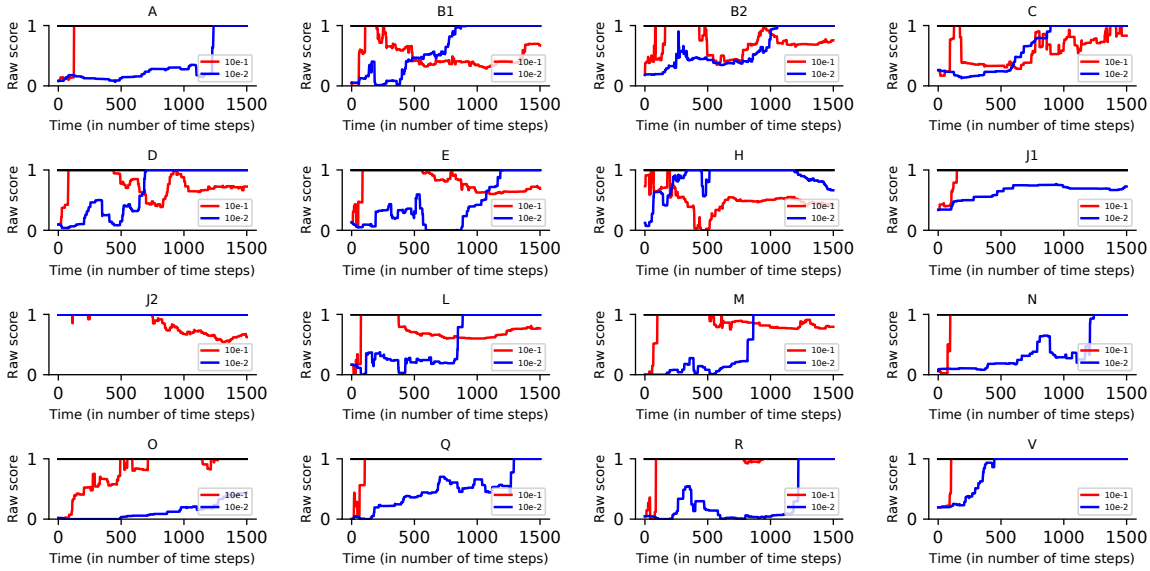


(b) Sensory response over time obtained from *classifier-EXT*.

Figure 5.5: **Learning rate effect on the softmax score**. Sensory response evolution for $\eta = 0.1$ (red lines) and $\eta = 0.01$ (blue lines) when either *classifier-REAL* (a) or *classifier-EXT* (b) is used as the first layer of the sensory response function. The sensory response remains generally low until it sharply increases until a value of 1. The stability drops after a certain time. A higher learning rate results in faster learning dynamics. Here, $n_A = 16$, $n_M = 3$, $W_{t_0} \in [-0.001, 0.001]$, $t_f = 1500$.



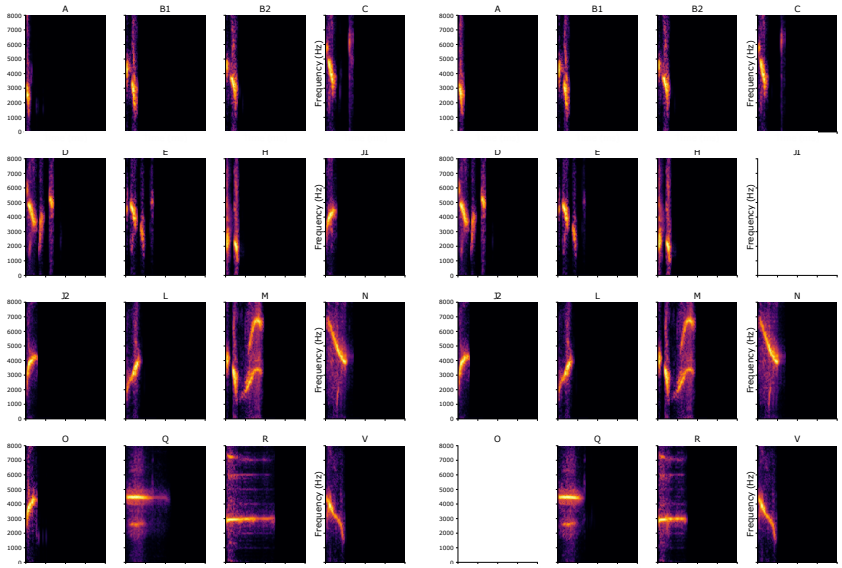
(a) Sensory response over time obtained from *classifier-REAL*.



(b) Sensory response over time obtained from *classifier-EXT*.

Figure 5.6: **Learning rate effect on the raw score.**. Sensory response evolution for $\eta = 0.1$ (red lines) and $\eta = 0.01$ (blue lines) when either *classifier-REAL* (a) or *classifier-EXT* (b) is used as sensory response function. For almost all the syllables, the sensory response gradually increases until a value of 1. Some syllables still show a sharp increase (e.g., syllable M in panel (a) or syllable R in panel (b)). The stability drops after a certain time. A higher learning rate results in faster learning dynamics. The sharpness is more pronounced for a bigger learning rate. Here, $n_A = 16$, $n_M = 3$, $W_{seed} = 0.001$, $t_f = 1500$ and *classifier-EXT* is used as the first layer of the sensory response function.

Such decay is due to the fact that the learning is driven by a simple Hebbian learning rule which is not expected to converge because it will never stop updating the weights (due to the absence of normalization). One can notice that a decay is also expected for those syllables that are stable at 1 at $t_f = 1500$: due to time constraints, it has not been possible to perform longer simulations.



(a) classifier-REAL (a) classifier-EXT

Figure 5.7: **Stability of syllable productions after learning**. Mean spectrogram obtained from all the sounds generated after learning (if a sound S produced at a certain time t_i is such that $raw\_score^S_{t_i} \geq 0.9$, then we consider all the sounds produced during learning from t_i and until t_f.). An empty box in panel (b) means that for that class, for all the produced sounds, we obtain $raw\_score^S_t < 0.9$, $forall S, \forall t_0 < t < t_f$. The learning rate is fixed at $\eta = 0.01$ and either *classifier-REAL* or *classifier-EXT* is used as the first layer of the sensory response function. Here, $n_A = 16$, $n_M = 3$, $W_{t_0} \in U[-0.001, 0.001]$, $t_f = 1500$. The raw score has been used to normalize the output of the classifier.

Alternatively, the usage of a normalized raw score as the one defined by Equation 5.1 results in a less abrupt trend (Figure 5.6) with respect to the dynamics observed in Figure 5.5. A sharp increase still happens for some syllables (e.g., syllable M in Figure 5.6(a) or syllable R in Figure 5.6(b)), but in general the sensory response evolution is more gradual. Even if not shown, a decay is expected for those syllables that are stable at 1 at

$t_f = 1500$. Due to time constraints, it has not been possible to perform simulations longer than 1500 time steps.

The sounds produced after learning (for each class of syllables, we consider the syllable as learned when the raw score is stably higher than 0.9) is stable for any class of syllables both when *classifier-REAL* (Figure 5.7a) and *classifier-EXT* (Figure 5.7b) is used as the first layer of the sensory response function. Empty boxes in Figure 5.7b mean that for syllable *O* and syllable *J1* the raw score is never higher than 0.9 (see Figure 5.6).

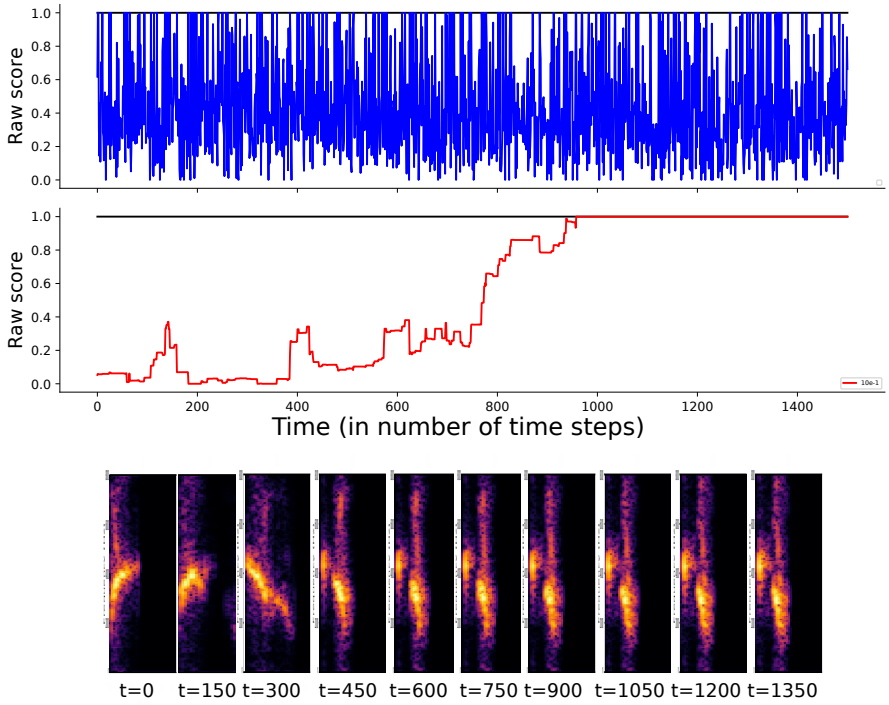


Figure 5.8: **Relation between the motor exploration and the sensory response: syllable *B1*.** Distribution of the sensory responses (raw score) obtained at each time t from the random motor exploration relative to syllable *B1* (blue line in the upper panel). Sensory response (raw score) evolution over time for $\eta = 0.01$ (red line in the middle panel). Evolution of the sensory production corresponding to the raw score every 150 time steps (bottom panel). Here, $n_A = 16$, $n_M = 3$, $W_{t_0} \in [-0.001, 0.001]$, $t_f = 1500$, and *classifier-REAL* is used in the first layer of the sensory response function. The raw score is used to normalize the output of the classifier.

During learning, a transition from one syllable to another in the sensory space occurs

in parallel to the raw score dynamics (Figures 5.8(b-c) and 5.9). Syllables *B1* and *C* have been chosen as representative elements to show such a transition when, respectively, *classifier-REAL* and *classifier-EXT* are used as first layer of the sensory response function.

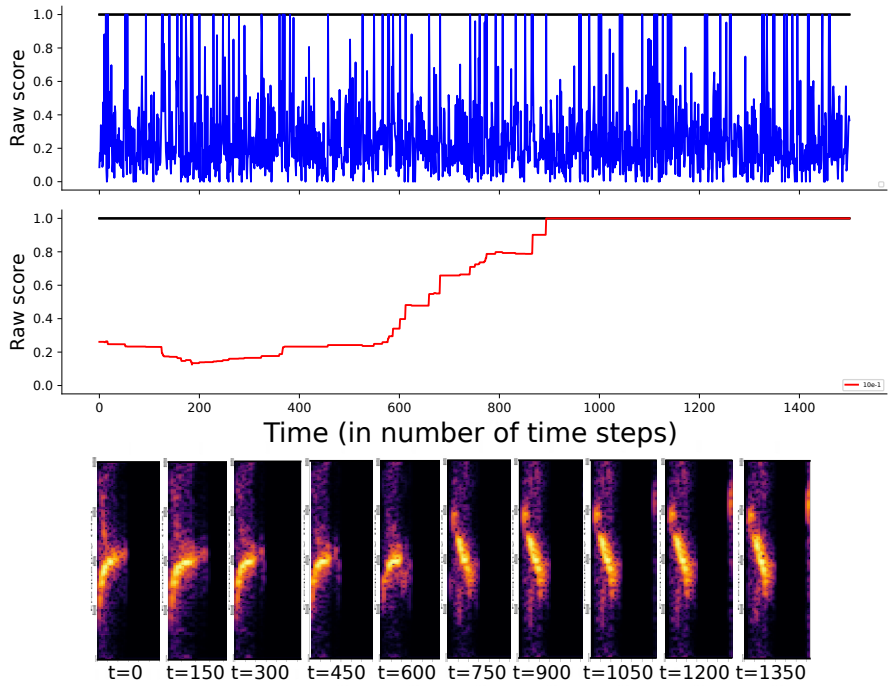


Figure 5.9: **Relation between the motor exploration and the sensory response: syllable *C*.** Distribution of the sensory responses (raw score) obtained at each time t from the random motor exploration relative to syllable C (blue line in the upper panel). Sensory response (raw score) evolution over time for $\eta = 0.01$ (red line in the middle panel). Evolution of the sensory production corresponding to the raw score every 150 time steps (bottom panel). Here, $n_A = 16$, $n_M = 3$, $W_{t_0} \in [-0.001, 0.001]$, $t_f = 1500$, and *classifier-EXT* is used in the first layer of the sensory response function. The raw score is used to normalize the output of the classifier.

5.4 Discussion

In this chapter, we built a vocal learning model with a full action-perception loop (as the model suggested by Figure 2.1 in Chapter 2). We aim the model to learn a repertoire of 16 different classes of canary syllables. The motor space is the 3-dimensional latent space

obtained from training WaveGAN (see Chapter 4). The motor control function is a generator model that enables the production of syllables resembling real recordings. Thus, the sensory space encodes the actual syllables produced. The sensory response function encodes the generated syllables in a rather low-dimensional space, i.e. the perceptual space. We used a classifier as the one introduced in Section 4.3.4 of Chapter 4 as sensory response function. The learning algorithm includes a random motor exploration strategy and a simple Hebbian learning rule: it drives the learning of the inverse model between the perceptual space and the motor space. We tested how the learning is influenced by (1) different learning rates, (2) different sensory response functions (*classifier-REAL* versus *classifier-EXT*), and (3) different sensory response definitions (softmax versus raw score). We showed that a simple Hebbian learning rule allows the learning but does not prevent divergence after a certain time step t_{critic}. The value of t_{critic} is syllable-specific (Figure 5.5) and depends on the learning rate. A higher learning rate results in faster learning dynamics and, thus, in an earlier t_{critic}. We compared two different ways to compute the sensory response: the usage of a softmax function (to obtain the probability distribution) may result in sharp transitions dynamics (see Figure 5.5), whereas a normalized raw score allows smoother transitions over time (see Figure 5.6).

Several modelers have proposed dynamic systems to model the motor control function in songbirds: in such case, the motor space describes the time-dependent motor articulations parameters which control the dynamics of the syrinx (e.g., air pressure, syringeal labial tension) (Doya and Sejnowski, 1998; Fiete et al., 2007; Amador et al., 2013; Alonso et al., 2015). Such dynamical systems have been used also in vocal development study to model the vocal production in marmoset (Teramoto et al., 2017).

As mentioned in Section 5.2.2, this brings to have a redundant motor space where multiple motor configurations correspond to the same syllable production (in terms of classes). Moreover, the motor space (properly called GAN *latent space*) has a high density of good productions (sounds highly activating one syllable in the perceptual layer).. On the one hand, the generator model learns well how to produce syllables resembling the training data (based on a clean dataset). On the other hand, the classifier has high

accuracy when it comes to assigning each syllable to a syllable class. This results almost always in a "good" assignment (to one class of the repertoire and not to an unknown class X). To force the motor space to be less dense of "good" syllables one could either (1) modify the training dataset by introducing artificially modified syllables (e.g., by adding some noise or syllable interpolations[1]) or (2) do not use the generator model at convergence but rather stop its training earlier (to have a higher percentage of "bad" productions). Further investigations of the motor space (latent space) are needed to understand (1) how it is structured and what is its topology (where each class of syllables is located in the 3-dimensional space), and (2) if the particular topology would give a hint for a particular exploration strategy.

A simple Hebbian learning rule allows the learning to reach the "targetted" perceptual goals but does not include stopping criteria. The introduction of a reinforcement signal could help the learning to stabilize after having reached the region of the motor space region that enables the production of the correct perceptual goal. The results obtained in this chapter aim to be expanded (1) by choosing a longer time t_f to stop the learning, (2) by exploring different strategies of exploration (e.g., goal-directed) and (3) by modifying the learning rule (e.g., combining Hebbian learning rule with Reinforcement Learning).

[1] Using interpolations of syllable generated from a previous generator model of a GAN.

Conclusions and perspectives

A deep understanding and comparison of the existing models in the literature of vocal learning has been simplified by using a common schema (Figures 2.1 and 2.2 of Chapter 2). The schema helps to unravel the models in their components and uncover the common structure they share. While the learning architecture remains an important component of a vocal learning model, the review of the literature highlighted how also the motor control model and the sensory response model play crucial roles in the definition of a complete vocal learning model. Having such a scenario in mind, and adding the knowledge of behavioral studies and neuroanatomical structure of the brain areas involved in vocal learning, it becomes easier to understand the objective of each study, the biological limitations, and the computational choices of each author.

Before approaching a complete vocal learning model including sound production, a simple theoretical model served as a first case study to understand how to build a bio-inspired vocal learning model. The model introduced in Chapter 3 does not enable sound production and is defined by a simple one-layer perceptron (Figure 3.1). On the one side, the motor area corresponds to the brain areas involved in the motor apparatus control. On the other side, the auditory area represents the brain areas where sensory stimuli are encoded. The theoretical inverse model learns the connections between the two populations driven by a normalized Hebbian learning rule. Such a normalization in the learning rule has been introduced to prevent the explosion of synaptic weights, i.e., to model the limit in the number of synapsis that a neuron can do in nature (Abbott and Nelson, 2000). Moreover, the introduction of a non-linear sensory response to the model

has been chosen to model sparse and nonlinear auditory response in the brain (Hahnloser and Kotowicz, 2010). The influence of the model parameters on the learning speed and accuracy highlights perspectives and limitations. On the one hand, the influence of the tuning selectivity on the learning speed and accuracy represents a trade-off that can be solved by introducing an evolutionary tuning selectivity (i.e., changing over time). On the other hand, the fact that the motor dimension drastically influences the learning speed points out a computational limitation of the model.

The definition of a complete vocal learning model presupposes the definition of a motor control model enabling sound production (Chapter 2). Chapter 4 draws on generative neural networks and proposed the generator part of WaveGAN (Donahue et al., 2018) (Generative Adversarial Network (GAN)) as a generative model for canary songs. The generator is capable of producing good outputs, i.e., resembling the training data (Figure 4.5) even when dealing with a low-dimensional latent space (Figure 4.7). Moreover, the 3-dimensional latent space used as input space for the generator turned out to be smooth when exploring the transition from one syllable to another (Figure 4.16). Chapter 4 also contains two salient methods. The proposed classifier can classify the real and the generated syllables as belonging to one particular class of the training data. The recently introduced representation method Uniform Manifold Approximation and Projection (UMAP) that allows the representation of a complex dataset in a two hyperparameter space where clusters and connections between clusters can be analyzed.

The fact that the generator part of WaveGAN works with a smooth low-dimensional input space suggests that it could be used as a motor control function in a vocal learning model. Chapter 5 introduces a vocal learning model with a full action-perception loop (Figure 2.1). That is a model containing a motor space, a sensory space, and a perceptual space connected by a motor control function, a sensory response function, and a learning architecture (Figure 5.1). The model aims to learn each of the syllables present in the target set independently. The motor control function is a generator model obtained from a 3-dimensional WaveGAN. The model *classifier-REAL* proposed in Chapter 4 is used as a sensory response: it provides the probability of each generation of belonging to each class

of the target set. The connections between the motor space and the perceptual space (represented by the probability vector described previously) are initially weak and the learning is driven by a simple Hebbian learning rule. As for the simple model described in Chapter 3, the motor space is uniformly explored and no goal-directed exploration takes place.

Positioning the work

The following paragraphs focus on the limitations, advantages, and perspectives of the approach presented in this thesis. Moreover, it places this thesis in the context of vocal learning in terms of goal, hypothesis and methods, and relates it to other published studies aiming at modeling vocal learning.

Motor control

Speech production models that aims to be bio-inspired take into account the anatomical configuration of the vocal production apparatus. For instance, models based on the structure of the vocal folds and the vocal tract (Titze, 1989; Titze and Martin, 1998; Fant, 2012). The former realizes the production of the sound thanks to the combination of the output of the vocal folds vibration and of the noise. While for birds the investigation is complicated given their small size, models able to reproduce vocal folds oscillations have been proposed for humans (Ishizaka and Flanagan, 1972; Birkholz, 2011; Amador et al., 2013; Mindlin, 2013). The vocal tract acts as a filter for the sound produced by the vocal folds: it modifies it by balancing its frequency components. The vocal tract, and in particular the air pressure dynamics, has been often modeled using Ordinary Differential Equations (ODEs) (Westerman and Miranda, 2002; Maeda, 1989). Several synthesizer attempts to model the non-linear structure of the vocal tract and introduces a high number of parameter to describe all the involved muscles (e.g., tongue, jaw). For instance, this is the case of Vocal Linear Articulatory Model (VLAM) (Maeda, 1990), Vocal Tract Lab

(VTL) (Birkholz et al., 2006; Birkholz, Accessed Sept. 2019) and Directions Into Velocities of Articulators (DIVA) (Guenther et al., 2006a; Tourville and Guenther, 2011). Similarly, the vocal tract dynamics in birds have been modeled using ODEs (Amador et al., 2013; Gardner et al., 2001). These models enable the production of sounds resembling the real ones for speech and, to a limited extent, birds. Moreover, they might be slow to simulate. For this reason, usually, only a few parameters influence the vocal production (they are time-dependent), while the remaining parameters are kept fix over time.

The brain seems unable to control each motor parameter independently but it uses a complex gesture-dependent control scheme to drive the vocal output (Elemans et al., 2015; Srivastava et al., 2015). There is a lack of experimental evidence regarding how the vocal production pathway develops in juvenile birds. At the beginning of their life, juveniles do not have developed motor control. Later, each RA stimulation enables the production of vocalizations in adult zebra finches and canaries (Vicario and Simpson, 1995).

Recently, artificial neural networks have been explored as a possible way to describe the non-linear multi-layer structure of the brain (Yamins et al., 2014; Ponce et al., 2019). In the original work from Donahue et al. (2018), the aim of the authors is to verify whether or not WaveGAN is able to reproduce realistic speech and birdsong (from wild recordings). In Chapter 4, WaveGAN has been trained on a dataset of recordings from an adult canary. The restriction of the dataset to only one canary reduces the variability with respect of using a set of wild recordings coming from different species. Moreover, it takes into account that, usually, canaries learn from their conspecific tutor. The goal of the thesis is to describe the sensorimotor phase of learning, which allows the bird to go from producing subsong to produce crystallized syllables (similar to the tutor's syllables). The training of WaveGAN has been done using adult recordings. Indeed, the vocal learning model should be able to learn adult songs (i.e., crystallized songs) and, in principle, a model trained on recordings from a different learning phase (e.g., subsongs from juveniles) would not be able to produce adult-like syllables. Moreover, juvenile' songs show a higher variability reflecting in a high variability. Whether or not WaveGAN can deal and converge with

juveniles data remains unknown. A key characteristic of WaveGAN, and of GANs in general, is the redundancy of the latent space. When using such a model as motor control function in a vocal learning model, this means that multiple motor configurations produce the same sensory output (i.e., the same sound). Regardless, a deeper comparison between a vocal learning model where the motor function is implemented using WaveGAN and one in which it is implemented using an ODE model based on the avian organ (e.g., the work from Amador et al. (2013)) remains needed.

Learning architecture

While reward-driven learning is associated with trial-and-error processes, associative learning is associated with the co-activation of neural populations. Hebbian learning is likely the mechanism for associative learning, while RL needs an additional reward signal, provided by neuromodulators.

Several modelers used a gradient-based RL algorithm to explain vocal learning in humans (Howard and Messum, 2007; Warlaumont and Finnegan, 2016; Howard and Birkholz, 2019) and birds (Doya and Sejnowski, 2000; Fiete et al., 2007; Troyer and Doupe, 2000). The hypothesis that the auditory feedback reinforces actions is tested by Howard and Messum (2007), Warlaumont and Finnegan (2016) and Howard and Birkholz (2019): the estimated auditory salience determines whether or not to reward the model. Alternatively, the existence of internal models is suggested by *one-shot learning* in humans. Hebbian-inspired learning rules have been proposed to drive learning when the learning architecture is based on internal models (Troyer and Doupe, 2000; Westerman and Miranda, 2002; Oudeyer, 2005; Kröger et al., 2009). Internal models (inverse and forward) usually modifies the simple Hebbian learning rule in order to make the model biologically plausible and allow convergence of the learning (Hahnloser and Ganguli, 2013).

A simple Hebbian learning rule can drive the learning at early stages but is not expected to converge. Alternatively, some normalization rules can assert convergence under certain conditions (Hahnloser and Ganguli, 2013). Similarly, the model proposed in Chap-

ter 3 defines an inverse model driven by a normalized Hebbian learning rule. Interestinlgy, a normalization over auditory neurons works better than a normalization over motor neurons: this might be connected to the fact that the target of the model is perceptual, thus corresponds to the *auditory dimension* of the synaptic weights. Chapter 3 provides a perspective on how the learning is influenced by the auditory selectivity and by the size of the network. A higher selectivity and a higher motor dimension introduce sparsity in the target, and thus makes it more difficult to learn.

Chapter 5 relies on a simple Hebbian learning rule to learn the connections between the perceptual and the motor space (see Figure 5.1) in a complete vocal learning model. A 3-dimensional generator model of WaveGAN is used as a motor control function. This means that (1) the model enables sound production and (2) the sound depends on 3 motor parameters. As mentioned above, such a motor control function leads to a redundant motor space. As a consequence, a reflection about how to define the target of the learning is needed. The solution proposed in Chapter 5 considers a perceptual target instead of a motor target. That is, each time a new sound is produced by the model and is processed by the sensory response function, it is awarded by a perceptual score. The model proposed in Chapter 5 can be extended and completed by introducing reinforcement learning (RL). The idea is to keep a simple Hebbian learning rule at the early stages of learning and introduce a goal-directed strategy when the perceptual target is closer. A combination of Hebbian learning and RL has been previously proposed by Troyer and Doupe (2000) and the perceptual score described above could be used to determine whether or not the model receives a reward or not (similarly to how auditory salience has been used with humans (Howard and Messum, 2007; Warlaumont and Finnegan, 2016; Howard and Birkholz, 2019).

Sensory response

The sensory response attempts to model how the sensory space (usually a sound in vocal learning) is perceived. The ability to perceive a continuous space (e.g. sound) as

discretized (e. g., phonemes) exists both in humans and birds (Kuhl, 2000, 2004). The auditory system enables the discrimination of songs, the recognition of the tutor, and the evaluation of auditory feedback (Hahnloser and Kotowicz, 2010). Moreover, in birds, neural selectivity develops during song ontogeny (Brainard and Doupe, 2002).

Computationally, the sensory response provides a perceptual representation of the sound. In reinforcement learning (RL), the sensory response may also trigger the reward. Usually, the sensory response function allows to the interpretation of the sound in a low-dimensional space thanks to a multi-step process that leads to the perceptual space representation. This is coherent with the evidence of highly non-linear and sparse responses found in the auditory pathway (Hahnloser and Kotowicz, 2010). For example, Philippsen et al. (2016) uses Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Alternatively, the sensory response function can be based on the sound features (Westerman and Miranda, 2002; Howard and Huckvale, 2005; Philippsen et al., 2014).

The model proposed in Chapter 3 models the auditory pathway proposing a non-linear auditory response similar to the one used by Westerman and Miranda (2002) and Oudeyer (2005). In particular, the sensory response defined in Chapter 3 does not rely on any feature of the sound since there is no sound production in the proposed model. The complete vocal learning model proposed in Chapter 4 introduces the generation of the sound and the sensory response to process the sound. The sensory response function is an Echo State Network (ESN), a type of artificial neural network designed to manipulate sequential data and appropriate to solve the problem of sound classification. The sensory response encodes the sound in the perceptual space as a probability vector which discriminates the class to which it belongs to.

Perspectives

As a general perspective on how this study can be extended, the following paragraphs explore how the proposed methods could be the subject of further investigation. The

classifier model proposed in Chapter 4 has been used to obtain a quantitative measure of the generated data. In this sense, the trade-off between having an accurate classifier and a too accurate classifier is a sensitive point. On the one hand, an accurate classifier is useful to distinguish the syllables from one another. This classification is needed to compare the syllables qualitatively and qualitatively[2]. On the other hand, a too accurate classifier could learn how to differentiate a training sample from a generated sample before learning how to classify them correctly. In this scenario, a good syllable could be classified as an alternative syllable (as it was defined in Section 4.3.3 of Chapter 4) because the classifier bases its decision on a binary choice real/generated. This was a sensible point in Chapter 4, and is the reason why we chose a classifier with lower accuracy on the real data because it was less prone to label generated syllables as "alternative syllables". Additionally, the classifier could help to detect errors in the selection of single syllables, and it could be useful to identify to which class a phrase belongs to by using the selection of the single syllables. Nevertheless, a percentage of manual work would still be needed to select the training labeled dataset.

The methods proposed in Chapter 4 to evaluate the generator model aim to describe the training and the generated data in a way they can be compared. To this end, we used qualitative measures such as the mean spectrogram and the UMAP representation. The latter serves as a bridge between a qualitative measure of the generated data and a representation of an exploration of the latent space. This is an important point both for the vocal learning model, when a low-dimensional motor space is required and for the motor control function itself, where continuity in the motor space is biologically plausible. To analyze better the capability of the models of reproducing good sounds, different motor control models should be tested and their output should be compared using a common representation.

WaveGAN, and in general generative models, could be able to serve as a motor control function for different vocal learning models. That is, one could use the same generator

[2]The classification provided by the classifier is needed to grup the syllables and represent them using, for example, the mean spectrogram or UMAP

(trained with different datasets) to model the motor control function in a vocal learning model trying to explain song learning in caries or speech development in humans. To test this possibility, the same generative model should be trained on different datasets to assess its capability of reproducing realistic outputs. Although there is the need to check the ability of each component of the model to deal with different data, potentially the model proposed in Chapter 5 can help explaining vocal learning in songbirds but open also to other perspectives. For instance, the same model structure could be used for vocal learning in humans or in artificial agents' communication.

Bibliography

L. Abbott and S. Nelson. Synaptic plasticity: taming the beast. *Nature neuroscience*, 3 (11s):1178, 2000.

J. Acevedo-Valle, V. Hafner, and C. Angulo. Social reinforcement in artificial prelinguistic development: A study using intrinsically motivated exploration architectures. *IEEE Transactions on Cognitive and Developmental Systems*, 2018.

R. Alonso, M. Trevisan, A. Amador, F. Goller, and G. Mindlin. A circular model for song motor control in serinus canaria. *Frontiers in computational neuroscience*, 9:41, 2015.

A. Amador, Y. Perl, G. Mindlin, and D. Margoliash. Elemental gesture dynamics are encoded by song premotor cortical neurons. *Nature*, 495(7439):59, 2013.

A. Andalman and M. Fee. A basal ganglia-forebrain circuit in the songbird biases motor output to avoid vocal errors. *PNAS*, 106(30):12518–12523, 2009.

M. Arbib. The mirror system, imitation, and the evolution of language. *Imitation in animals and artifacts*, 229, 2002.

M. A. Arbib. From monkey-like action recognition to human language: An evolutionary framework for neurolinguistics. *Behavioral and brain sciences*, 28(2):105–124, 2005.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

D. Aronov, A. S. Andalman, and M. S. Fee. A specialized forebrain circuit for vocal babbling in the juvenile songbird. *Science*, 320(5876):630–634, 2008.

G. Arriaga, E. P. Zhou, and E. D. Jarvis. Of mice, birds, and men: the mouse ultrasonic song system has some features similar to humans and song-learning birds. *PloS one*, 7 (10):e46610, 2012.

R. Artstein and M. Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008.

G. Bailly. Learning to speak. sensori-motor control of speech movements. *Speech Communication*, 22(2-3):251–267, 1997.

A. Baranes and P. Oudeyer. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, 2013.

M. Barnaud, J. Schwartz, P. Bessière, and J. Diard. Computer simulations of coupled idiosyncrasies in speech perception and speech production with cosmo, a perceptuo-motor bayesian model of speech communication. *PloS one*, 14(1):e0210302, 2019.

S. Barratt and R. Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

N. Baudonck, R. Buekers, S. Gillebert, and K. Van Lierde. Speech intelligibility of flemish children as judged by their parents. *Folia Phoniatrica et Logopaedica*, 61(5):288–295, 2009.

G. J. Beckers, J. J. Bolhuis, K. Okanoya, and R. C. Berwick. Birdsong neurolinguistics: Songbird context-free grammar claim is premature. *Neuroreport*, 23(3):139–145, 2012.

P. Bédard and J. N. Sanes. Basal ganglia-dependent processes in recalling learned visual-motor adaptations. *Experimental brain research*, 209(3):385–393, 2011.

S. Belzner, C. Voigt, C. K. Catchpole, and S. Leitner. Song learning in domesticated canaries in a restricted acoustic environment. *Proceedings of the Royal Society B: Biological Sciences*, 276(1669):2881–2886, 2009.

D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

P. Birkholz. A survey of self-oscillating lumped-element models of the vocal folds. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2011*, pages 47–58, 2011.

P. Birkholz. Vocaltractlab–towards high-quality articulatory speech synthesis. *http://www.vocaltractlab.de/*, Accessed Sept. 2019.

P. Birkholz, D. Jackèl, and B. Kroger. Construction and control of a three-dimensional vocal tract model. In *ICASSP*, volume 1. IEEE, 2006.

S. Boari, Y. Perl, . Amador, . Margoliash, and . Mindlin. Automatic reconstruction of physiological gestures used in a model of birdsong production. *Journal of neurophysiology*, 114(5):2912–2922, 2015.

P. Boersma et al. *Functional phonology: Formalizing the interactions between articulatory and perceptual drives*, volume 11. Holland Academic Graphics The Hague, 1998.

C. Boettiger and A. Doupe. Developmentally restricted synaptic plasticity in a songbird nucleus required for song learning. *Neuron*, 31(5):809–818, 2001.

A. Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

K. E. Bouchard and M. S. Brainard. Neural encoding and integration of learned probabilistic sequences in avian sensory-motor circuitry. *Journal of Neuroscience*, 33(45): 17710–17723, 2013.

K. E. Bouchard and M. S. Brainard. Auditory-induced neural dynamics in sensory-motor circuitry predict learned temporal and sequential statistics of birdsong. *Proceedings of the National Academy of Sciences*, 113(34):9641–9646, 2016.

K. E. Bouchard, N. Mesgarani, K. Johnson, and E. F. Chang. Functional organization of human sensorimotor cortex for speech articulation. *Nature*, 495(7441):327–332, 2013.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

M. Brainard and A. Doupe. What songbirds teach us about learning. *Nature*, 417(6886): 351, 2002.

E. A. Brenowitz and M. D. Beecher. Song learning in birds: diversity and plasticity, opportunities and challenges. *Trends in neurosciences*, 28(3):127–132, 2005.

E. H. Buder, A. S. Warlaumont, and D. K. Oller. An acoustic phonetic catalog of prespeech vocalizations from a developmental perspective. *Comprehensive perspectives on child speech development and disorders: Pathways from linguistic theory to clinical practice*, 4:103–134, 2013.

T. A. Burnett, M. B. Freedland, C. R. Larson, and T. C. Hain. Voice f0 responses to manipulations in pitch feedback. *The Journal of the Acoustical Society of America*, 103 (6):3153–3161, 1998.

M. Chakraborty and E. Jarvis. Brain evolution by brain pathway duplication. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1684):20150056, 2015.

J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

L. Cohen and A. Billard. Social babbling: The emergence of symbolic gestures and words. *Neural Networks*, 2018.

K. A. Cross and M. Iacoboni. Neural systems for preparatory control of imitation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1644):20130176, 2014.

R. Darshan, W. Wood, S. Peters, A. Leblois, and D. Hansel. A canonical neural mechanism for behavioral variability. *Nature communications*, 8:15415, 2017.

B. De Boer. Self-organization in vowel systems. *Journal of phonetics*, 28(4):441–465, 2000.

B. De Boer. *The origins of vowel systems*, volume 1. Oxford University Press on Demand, 2001.

E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751*, 2015.

A. Dhawale, M. Smith, and B. Ölveczky. The role of variability in motor learning. *Annual review of neuroscience*, 40:479–498, 2017.

G. Di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: a neurophysiological study. *Experimental brain research*, 91(1):176–180, 1992.

L. Ding and D. Perkel. Long-term potentiation in an avian basal ganglia nucleus essential for vocal learning. *Journal of Neuroscience*, 24(2):488–494, 2004.

C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

C. Donahue, J. McAuley, and M. Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

A. Doupe and P. Kuhl. Birdsong and human speech: common themes and mechanisms. *Annual review of neuroscience*, 22(1):567–631, 1999.

A. Doupe, D. Perkel, A. Reiner, and E. Stern. Birdbrains could teach basal ganglia research a new song. *Trends in neurosciences*, 28(7):353–363, 2005.

A. J. Doupe. Song-and order-selective neurons in the songbird anterior forebrain and their emergence during vocal development. *Journal of Neuroscience*, 17(3):1147–1167, 1997.

K. Doya. Complementary roles of basal ganglia and cerebellum in learning and motor control. *Current opinion in neurobiology*, 10(6):732–739, 2000.

K. Doya and T. Sejnowski. A computational model of birdsong learning by auditory experience and auditory feedback. In *Central auditory processing and neural modeling*, pages 77–88. Springer, 1998.

K. Doya and T. Sejnowski. A computational model of avian song learning. In *The new cognitive neurosciences (2nd ed.) Gazzaniga, M. S. (Ed.)*. Cambridge, MA, US: The MIT Press., 2000.

D. Düring, B. Knörlein, and C. Elemans. In situ vocal fold properties and pitch prediction by dynamic actuation of the songbird syrinx. *Scientific reports*, 7(1):11296, 2017.

C. Elemans, J. Rasmussen, C. Herbst, D. Düring, S. Zollinger, H. Brumm, K. Srivastava, N. Svane, M. Ding, O. Larsen, et al. Universal mechanisms of sound production and control in birds and mammals. *Nature communications*, 6:8978, 2015.

J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*, pages 1068–1077. PMLR, 2017.

B. Erath, M. Zanartu, K. Stewart, M. Plesniak, D. Sommer, and S. Peterson. A review of lumped-element models of voiced speech. *Speech Communication*, 55(5):667–690, 2013.

G. Fant. *Acoustic theory of speech production: with calculations based on X-ray studies of Russian articulations*, volume 2. Walter de Gruyter, 2012.

P. F. Ferrari and G. Rizzolatti. *New frontiers in mirror neurons research*. Oxford University Press, USA, 2015.

P. F. Ferrari, V. Gallese, G. Rizzolatti, and L. Fogassi. Mirror neurons responding to the observation of ingestive and communicative mouth actions in the monkey ventral premotor cortex. *European journal of neuroscience*, 17(8):1703–1714, 2003.

I. Fiete, M. Fee, and H. Seung. Model of birdsong learning based on gradient estimation by dynamic perturbation of neural conductances. *Journal of neurophysiology*, 98(4): 2038–2057, 2007.

S. Forestier and P. Oudeyer. Curiosity-driven development of tool use precursors: a computational model. In *CogSci 2016*, pages 1859–1864, 2016.

S. Forestier and P. Oudeyer. A unified model of speech and tool use early development. In *CogSci 2017*, 2017.

S. Forestier, Y. Mollard, and P. Oudeyer. Intrinsically motivated goal exploration processes with automatic curriculum learning. *arXiv preprint arXiv:1708.02190*, 2017.

F. S. Foundation. http://espeak.sourceforge.net/.accessed 2011 dec 13. 2007.

C. Fowler. Speech perception as a perceptuo-motor skill. In *Neurobiology of Language*, pages 175–184. Elsevier, 2016.

V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

A. Friederici. The brain basis of language processing: from structure to function. *Physiological reviews*, 91(4):1357–1392, 2011.

J. M. Galea, A. Vazquez, N. Pasricha, J.-J. Orban de Xivry, and P. Celnik. Dissociating the roles of the cerebellum and motor cortex during adaptive learning: the motor cortex retains what the cerebellum learns. *Cerebral cortex*, 21(8):1761–1770, 2011.

V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119(2):593–609, 1996.

T. Gardner, G. Cecchi, M. Magnasco, R. Laje, and G. B. Mindlin. Simple motor gestures for birdsongs. *Physical review letters*, 87(20):208101, 2001.

R. Gentner and J. Classen. Modular organization of finger movements by the human central nervous system. *Neuron*, 52(4):731–742, 2006.

A. Ghazanfar and D. Liao. Constraints and flexibility during vocal development: insights from marmoset monkeys. *Current opinion in behavioral sciences*, 21:27–32, 2018.

N. Giret, J. Kornfeld, S. Ganguli, and R. H. Hahnloser. Evidence for a causal inverse model in an avian cortico-basal ganglia circuit. *PNAS*, 111(16):6063–6068, 2014.

J. Goldberg, M. Farries, and M. Fee. Basal ganglia output to the thalamus: still a paradox. *Trends in neurosciences*, 36(12):695–705, 2013.

I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

A. M. Graybiel. The basal ganglia: learning new tricks and loving it. *Current opinion in neurobiology*, 15(6):638–644, 2005.

D. Griffin and J. Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

J. Gudhnason, D. Mehta, and T. Quatieri. Evaluation of speech inverse filtering techniques using a physiologically based synthesizer. In *ICASSP*, pages 4245–4249. IEEE, 2015.

F. Guenther, S. Ghosh, A. Nieto-Castanon, and J. Tourville. A neural model of speech production. *Speech production: Models, phonetic processes and techniques*, pages 27–40, 2006a.

F. H. Guenther, S. S. Ghosh, and J. A. Tourville. Neural modeling and imaging of the cortical interactions underlying syllable production. *Brain and language*, 96(3):280–301, 2006b.

J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937*, 2020.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

H. R. Güttinger. Consequences of domestication on the song structures in the canary. *Behaviour*, 94(3-4):254–278, 1985.

H. R. Güttinger, W. Jochen, and T. Franz. The relationship between species specific song programs and individual learning in songbirds. *Behaviour*, 65(1-2):241–261, 1978.

D. Ha and D. Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

R. Hahnloser and S. Ganguli. Vocal learning with inverse models. *Principles of Neural Coding*, pages 547–564, 2013.

R. H. Hahnloser and A. Kotowicz. Auditory representations and memory in birdsong learning. *Current opinion in neurobiology*, 20(3):332–339, 2010.

R. H. Hahnloser and G. Narula. A bayesian account of vocal adaptation to pitch-shifted auditory feedback. *PloS one*, 12(1):e0169795, 2017.

A. Hanuschkin, S. Ganguli, and R. H. Hahnloser. A hebbian learning rule gives rise to mirror neurons and links them to control theoretic inverse models. *Frontiers in neural circuits*, 7:106, 2013.

K. J. Hayes and C. Hayes. Imitation in a home-raised chimpanzee. *Journal of comparative and physiological psychology*, 45(5):450, 1952.

C. Heyes. Causes and consequences of imitation. *Trends in cognitive sciences*, 5(6): 253–261, 2001.

C. Heyes. Where do mirror neurons come from? *Neuroscience & Biobehavioral Reviews*, 34(4):575–583, 2010.

C. Heyes. What's social about social learning? *Journal of Comparative Psychology*, 126 (2):193, 2012.

G. Hickok and D. Poeppel. The cortical organization of speech processing. *Nature reviews neuroscience*, 8(5):393–402, 2007.

G. Hickok, J. Houde, and F. Rong. Sensorimotor integration in speech processing: computational basis and neural organization. *Neuron*, 69(3):407–422, 2011.

J. Hillenbrand, L. A. Getty, M. J. Clark, and K. Wheeler. Acoustic characteristics of american english vowels. *The Journal of the Acoustical society of America*, 97(5):3099–3111, 1995.

X. Hinaut and P. F. Dominey. Real-time parallel processing of grammatical structure in the fronto-striatal system: A recurrent network simulation study using reservoir computing. *PloS one*, 8(2):e52946, 2013.

I. Howard and P. Birkholz. Modelling vowel acquisition using the birkholz synthesizer. *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*, pages 304–311, 2019.

I. Howard and M. Huckvale. Training a vocal tract synthesiser to imitate speech using distal supervised learning. In *SPECOM*, volume 2, pages 159–162. University of Patras, Wire Communications Laboratory, 2005.

I. Howard and P. Messum. A computational model of infant speech development. In *SPECOM*, pages 756–765, 2007.

I. Howard and P. Messum. Modeling the development of pronunciation in infant speech acquisition. *Motor Control*, 15(1):85–117, 2011.

M. Iacoboni, R. P. Woods, M. Brass, H. Bekkering, J. C. Mazziotta, and G. Rizzolatti. Cortical mechanisms of human imitation. *science*, 286(5449):2526–2528, 1999.

K. Ishizaka and J. Flanagan. Synthesis of voiced sounds from a two-mass model of the vocal cords. *Bell system technical journal*, 51(6):1233–1268, 1972.

H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

V. M. Janik and P. J. Slater. Vocal learning in mammals. *Advances in the Study of Behaviour*, 26:59–100, 1997.

E. Jarvis. Evolution of vocal learning and spoken language. *Science*, 366(6461):50–54, 2019.

M. Jordan and D. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive science*, 16(3):307–354, 1992.

M. Jueptner, C. Frith, D. Brooks, R. Frackowiak, and R. Passingham. Anatomy of motor learning. ii. subcortical structures and learning by trial and error. *Journal of neurophysiology*, 77(3):1325–1337, 1997.

M. Kawato. Feedback-error-learning neural network for supervised motor learning. In *Advanced neural computers*, pages 365–372. Elsevier, 1990.

M. Kawato. Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727, 1999.

G. Keller and R. H. Hahnloser. Neural processing of auditory feedback during vocal practice in a songbird. *Nature*, 457(7226):187, 2009.

R. D. Kent and A. D. Murray. Acoustic features of infant vocalic utterances at 3, 6, and 9 months. *The Journal of the Acoustical Society of America*, 72(2):353–365, 1982.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

J. W. Krakauer and P. Mazzoni. Human sensorimotor learning: adaptation, skill, and beyond. *Current opinion in neurobiology*, 21(4):636–644, 2011.

B. Kröger, J. Kannampuzha, and C. Neuschaefer-Rube. Towards a neurocomputational model of speech production and perception. *Speech Communication*, 51(9):793–809, 2009.

P. Kuhl. A new view of language acquisition. *Proceedings of the National Academy of Sciences*, 97(22):11850–11857, 2000.

P. Kuhl. Early language acquisition: cracking the speech code. *Nature reviews neuroscience*, 5(11):831, 2004.

P. Ladefoged. *Elements of acoustic phonetics*. University of Chicago Press, 1996.

A. Laversanne-Finot, A. Péré, and P. Oudeyer. Curiosity driven exploration of learned disentangled goal spaces. *arXiv preprint arXiv:1807.01521*, 2018.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

R. Legenstein, S. Chase, A. Schwartz, and W. Maass. A reward-modulated hebbian learning rule can explain experimentally observed network reorganization in a brain control task. *Journal of Neuroscience*, 30(25):8400–8410, June 2010.

K. Lehongre, T. Aubin, S. Robin, and C. Del Negro. Individual signature in canary songs: contribution of multiple levels of song structure. *Ethology*, 114(5):425–435, 2008.

A. Leonardo and M. S. Fee. Ensemble coding of vocal control in birdsong. *Journal of Neuroscience*, 25(3):652–661, 2005.

X. Li and X. Wu. Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4520–4524. IEEE, 2015.

A. Liberman and I. Mattingly. The motor theory of speech perception revised. *Cognition*, 21(1):1–36, 1985.

A. M. Liberman, F. S. Cooper, D. P. Shankweiler, and M. Studdert-Kennedy. Perception of the speech code. *Psychological review*, 74(6):431, 1967.

J. Liljencrants, B. Lindblom, et al. Numerical simulation of vowel quality systems: The role of perceptual contrast. *Language*, 48(4):839–862, 1972.

H. Liu and Y. Xu. Learning model-based f0 production through goal-directed babbling. In *ISCSLP*, pages 284–288. IEEE, 2014.

A. J. Lotto, G. S. Hickok, and L. L. Holt. Reflections on mirror neurons and speech perception. *Trends in cognitive sciences*, 13(3):110–114, 2009.

C. Lyon, C. Nehaniv, and J. Saunders. Interactive language learning by robots: The transition from babbling to word forms. *PloS one*, 7(6):e38236, 2012.

S. Maeda. Vtcalcs. *http://ed268. univ-paris3. fr/lpp/index. php? page= ressources/logiciels.*

S. Maeda. Compensatory articulation in speech: analysis of x-ray data with an articulatory model. In *First European Conference on Speech Communication and Technology*, 1989.

S. Maeda. Compensatory articulation during speech: Evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model. In *Speech production and speech modelling*, pages 131–149. Springer, 1990.

J. E. Markowitz, E. Ivie, L. Kligler, and T. J. Gardner. Long-range order in canary song. *PLoS Comput Biol*, 9(5):e1003052, 2013.

P. Mazzoni and J. W. Krakauer. An implicit plan overrides an explicit strategy during visuomotor adaptation. *Journal of neuroscience*, 26(14):3642–3645, 2006.

L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

W. Mehaffey and A. Doupe. Naturalistic stimulation drives opposing heterosynaptic plasticity at two inputs to songbird cortex. *Nature neuroscience*, 18(9):1272, 2015.

L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

J. L. Miller and A. M. Liberman. Some effects of later-occurring information on the perception of stop consonant and semivowel. *Perception & Psychophysics*, 25(6):457–465, 1979.

G. Mindlin. The physics of birdsong production. *Contemporary physics*, 54(2):91–96, 2013.

R. Mooney. Neural mechanisms for learned birdsong. *Learning & Memory*, 16(11):655–669, 2009.

C. L. Morgan. An introduction to comparative psychology, new ed., rev. 1903.

C. Moulin-Frier and P. Oudeyer. Curiosity-driven phonetic learning. In *ICDL-EpiRob*, pages 1–8. IEEE, 2012.

C. Moulin-Frier, S. Nguyen, and P. Oudeyer. Self-organization of early vocal development in infants and machines: the role of intrinsic motivation. *Frontiers in psychology*, 4: 1006, 2014.

C. Moulin-Frier, J. Diard, J. Schwartz, and P. Bessière. 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.

M. Murakami, B. Kröger, P. Birkholz, and J. Triesch. Seeing [u] aids vocal learning: Babbling and imitation of vowels using a 3d vocal tract model, reinforcement learning, and reservoir computing. In *ICDL-EpiRob*, pages 208–213. IEEE, 2015.

S. Najnin and B. Banerjee. A predictive coding framework for a developmental agent: Speech motor skill acquisition and speech production. *Speech Communication*, 92:24–41, 2017.

A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

A. Y. Ng, S. J. Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

D. Oller. *The emergence of the speech capacity*. Psychology Press, 2000.

D. K. Oller. Infant vocalization and the development of speech. *Allied Health and Behavioral Sciences*, 1(4):523–549, 1978.

D. K. Oller and R. E. Eilers. The role of audition in infant babbling. *Child development*, pages 441–449, 1988.

D. K. Oller, E. H. Buder, H. L. Ramsdell, A. S. Warlaumont, L. Chorna, and R. Bakeman. Functional flexibility of infant vocalization and the emergence of language. *Proceedings of the National Academy of Sciences*, 110(16):6318–6323, 2013.

D. K. Oller, M. Caskey, H. Yoo, E. R. Bene, Y. Jhang, C.-C. Lee, D. D. Bowman, H. L. Long, E. H. Buder, and B. Vohr. Preterm and full term infant vocalization and the origin of language. *Scientific reports*, 9(1):1–10, 2019.

A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.

A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016b.

P. Oudeyer. The self-organization of speech sounds. *Journal of Theoretical Biology*, 233 (3):435–449, 2005.

P. Oudeyer, F. Kaplan, and V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

P.-Y. Oudeyer and F. Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

E. Oztop, M. Kawato, and M. Arbib. Mirror neurons and imitation: A computationally guided review. *Neural Networks*, 19(3):254–271, 2006.

E. Oztop, M. Kawato, and M. Arbib. Mirror neurons: functions, mechanisms and models. *Neuroscience letters*, 540:43–55, 2013.

S. Pagliarini, X. Hinaut, and A. Leblois. A bio-inspired model towards vocal gesture learning in songbird. In *ICDL Epirob, 2018*. IEEE, 2018a.

S. Pagliarini, A. Leblois, and X. Hinaut. Towards biological plausibility of vocal learning models: a short review. 2018b.

S. Pagliarini, A. Leblois, and X. Hinaut. Vocal imitation in sensorimotor learning models: a comparative review. *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

S. Pascual, A. Bonafonte, and J. Serra. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

C. I. Petkov and E. Jarvis. Birds, primates, and spoken language origins: behavioral phenotypes and neurobiological substrates. *Frontiers in evolutionary neuroscience*, 4: 12, 2012.

H. Petzka, A. Fischer, and D. Lukovnicov. On the regularization of wasserstein gans. *arXiv preprint arXiv:1709.08894*, 2017.

A. Philippsen. Goal-directed exploration for learning vowels and syllables: A computational model of speech acquisition. *KI-Künstliche Intelligenz*, pages 1–18, 2021.

A. Philippsen, R. Reinhart, and B. Wrede. Learning how to speak: Imitation-based refinement of syllable production in an articulatory-acoustic model. In *ICDL-EpiRob*, pages 195–200. IEEE, 2014.

A. Philippsen, R. Reinhart, and B. Wrede. Goal babbling of acoustic-articulatory models with adaptive exploration noise. In *ICDL-EpiRob*, pages 72–78. IEEE, 2016.

M. Pickering and S. Garrod. An integrated theory of language production and comprehension. *Behavioral and brain sciences*, 36(4):329–347, 2013.

C. R. Ponce, W. Xiao, P. F. Schade, T. S. Hartmann, G. Kreiman, and M. S. Livingstone. Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences. *Cell*, 177(4):999–1009, 2019.

J. Prather, S. Peters, S. Nowicki, and R. Mooney. Precise auditory–vocal mirroring in neurons for learned vocal communication. *Nature*, 451(7176):305, 2008.

S. Prom-on, P. Birkholz, and Y. Xu. Training an articulatory synthesizer with continuous acoustic data. In *INTERSPEECH*, pages 349–353, 2013.

F. Pulvermüller and L. Fadiga. Active perception: sensorimotor circuits as a cortical basis for language. *Nature reviews neuroscience*, 11(5):351–360, 2010.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

C. Raffel. Lakh midi dataset (lmd). *http://colinraffel.com/projects/lmd*, 2016.

R. Reinhart. *Reservoir computing with output feedback*. PhD Thesis. Bielefeld University, Germany, 2011.

R. F. Reinhart. Autonomous exploration of motor skills by skill babbling. *Autonomous Robots*, 41(7):1521–1537, 2017.

J. Reis, H. M. Schambra, L. G. Cohen, E. R. Buch, B. Fritsch, E. Zarahn, P. A. Celnik, and J. W. Krakauer. Noninvasive cortical stimulation enhances motor skill acquisition over multiple days through an effect on consolidation. *Proceedings of the National Academy of Sciences*, 106(5):1590–1595, 2009.

G. Rizzolatti and L. Craighero. The mirror-neuron system. *Annu. Rev. Neurosci.*, 27: 169–192, 2004.

G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi. Premotor cortex and the recognition of motor actions. *Cognitive brain research*, 3(2):131–141, 1996.

M. P. Robb, H. R. Bauer, and A. A. Tyler. A quantitative analysis of the single-word stage. *First Language*, 14(42-43):037–48, 1994.

A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck. A hierarchical latent vector model for learning long-term structure in music. *arXiv preprint arXiv:1803.05428*, 2018.

T. F. Roberts, S. M. Gobes, M. Murugan, B. P. Ölveczky, and R. Mooney. Motor circuits are required to encode a sensory model for imitative learning. *Nature neuroscience*, 15 (10):1454–1459, 2012.

M. Rohde, K. Narioka, J. J. Steil, L. K. Klein, and M. O. Ernst. Goal-related feedback guides motor exploration and redundancy resolution in human motor skill acquisition. *PLoS computational biology*, 15(3):e1006676, 2019.

M. Rolf. Goal babbling with unknown ranges: A direction-sampling approach. In *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–7. IEEE, 2013.

M. Rolf, J. Steil, and M. Gienger. Goal babbling permits direct learning of inverse kinematics. *IEEE Transactions on Autonomous Mental Development*, 2(3):216–229, 2010.

T. Sainburg, M. Thielk, and T. Q. Gentner. Latent space visualization, characterization, and generation of diverse vocal communication signals. *bioRxiv*, page 870311, 2019.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

C. Scharff and F. Nottebohm. A comparative study of the behavioral deficits following lesions of various parts of the zebra finch song system: implications for vocal learning. *Journal of Neuroscience*, 11(9):2896–2913, 1991.

W. Schultz. Predictive reward signal of dopamine neurons. *Journal of neurophysiology*, 80(1):1–27, 1998.

J.-L. Schwartz, A. Basirat, L. Ménard, and M. Sato. The perception-for-action-control theory (pact): A perceptuo-motor theory of speech perception. *Journal of Neurolinguistics*, 25(5):336–354, 2012.

T. Sejnowski. Storing covariance with nonlinearly interacting neurons. *Journal of mathematical biology*, 4(4):303–321, 1977.

R. M. Seyfarth, D. L. Cheney, and P. Marler. Monkey responses to three different alarm calls: evidence of predator classification and semantic communication. *Science*, 210 (4471):801–803, 1980.

R. Shadmehr, M. A. Smith, and J. W. Krakauer. Error correction, sensory prediction, and adaptation in motor control. *Annual review of neuroscience*, 33:89–108, 2010.

M. Sizemore and D. Perkel. Premotor synaptic plasticity limited to the critical period for song learning. *Proceedings of the National Academy of Sciences*, 108(42):17492–17497, 2011.

S. Sober, M. Wohlgemuth, and M. Brainard. Central contributions to acoustic variation in birdsong. *Journal of Neuroscience*, 28(41):10370–10379, 2008.

M. M. Solis and A. J. Doupe. Anterior forebrain neurons develop selectivity by an intermediate stage of birdsong learning. *Journal of Neuroscience*, 17(16):6447–6462, 1997.

K. Srivastava, C. Elemans, and S. Sober. Multifunctional and context-dependent control of vocal acoustics by individual muscles. *Journal of Neuroscience*, 35(42):14183–14194, 2015.

A. Stuart, J. Kalinowski, M. P. Rastatter, and K. Lynch. Effect of delayed auditory feedback on normal speakers at two speech rates. *The Journal of the Acoustical Society of America*, 111(5):2237–2241, 2002.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

H. Tanaka, T. J. Sejnowski, and J. W. Krakauer. Adaptation to visuomotor rotation through interaction between posterior parietal and motor cortical areas. *Journal of neurophysiology*, 102(5):2921–2932, 2009.

Y. Teramoto, D. Takahashi, P. Holmes, and A. Ghazanfar. Vocal development in a waddington landscape. *eLife*, 6:e20782, 2017.

L. Theis, A. v. d. Oord, and M. Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

F. Theunissen, K. Sen, and A. Doupe. Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *Journal of Neuroscience*, 20(6):2315–2331, 2000.

E. L. Thorndike. Animal intelligence: an experimental study of the associative processes in animals. *The Psychological Review: Monograph Supplements*, 2(4):i, 1898.

K. A. Thoroughman and R. Shadmehr. Learning of action through adaptive combination of motor primitives. *Nature*, 407(6805):742–747, 2000.

I. Titze. A four-parameter model of the glottis and vocal fold contact area. *Speech Communication*, 8(3):191–201, 1989.

I. Titze and D. Martin. Principles of voice production, 1998.

J. Tourville and F. Guenther. The diva model: A neural theory of speech acquisition and production. *Language and cognitive processes*, 26(7):952–981, 2011.

A. Tramacere and R. Moore. Reconsidering the role of manual imitation in language evolution. *Topoi*, 37(2):319–328, 2018.

A. Tramacere, K. Wada, K. Okanoya, A. Iriki, and P. Ferrari. Auditory-motor matching in vocal recognition and imitative learning. *Neuroscience*, 2019.

N. Trouvain, L. Pedrelli, T. T. Dinh, and X. Hinaut. Reservoirpy: an efficient and user-friendly library to design echo state networks. In *International Conference on Artificial Neural Networks*, pages 494–505. Springer, 2020.

T. Troyer and A. Doupe. An associational model of birdsong sensorimotor learning i. efference copy and the learning of song syllables. *Journal of Neurophysiology*, 84(3): 1204–1223, 2000.

P. L. Tyack. A taxonomy for vocal learning. *Philosophical Transactions of the Royal Society B*, 375(1789):20180406, 2020.

T. Verstynen and P. N. Sabes. How each movement changes the next: an experimental and theoretical study of fast adaptive priors in reaching. *Journal of Neuroscience*, 31 (27):10050–10059, 2011.

D. S. Vicario and H. B. Simpson. Electrical stimulation in forebrain nuclei elicits learned vocal patterns in songbirds. *Journal of neurophysiology*, 73(6):2602–2607, 1995.

R. S. Waldstein. Effects of postlingual deafness on speech production: implications for the role of auditory feedback. *The Journal of the Acoustical Society of America*, 88(5): 2099–2114, 1990.

A. Warlaumont. *The Cambridge Handbook of Infant Development: Brain, Behavior, and Cultural Context*, chapter Infant vocal learning and speech production., pages 602–631. Cambridge University Press, 2020.

A. Warlaumont and M. Finnegan. Learning to produce syllabic speech sounds via reward-modulated neural plasticity. *PloS one*, 11(1):e0145096, 2016.

G. Westerman and E. Miranda. Modelling the development of mirror neurons for auditory-motor integration. *Journal of new music research*, 31(4):367–375, 2002.

S. Wilson, A. Pinar Saygin, M. Sereno, and M. Iacoboni. Listening to speech activates motor areas involved in speech production. *Nature Neuroscience*, 7(7):701–702, June 2004.

J. D. Wittenbach, K. E. Bouchard, M. S. Brainard, and D. Z. Jin. An adapting auditory-motor feedback loop can contribute to generating vocal repetition. *PLoS Comput Biol*, 11(10):e1004471, 2015.

D. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11(7-8):1317–1329, 1998.

D. Wolpert, Z. Ghahramani, and J. Flanagan. Perspectives and problems in motor learning. *Trends in Cognitive Sciences*, 5(11):487–494, Nov. 2001.

D. Wolpert, J. Diedrichsen, and J. Flanagan. Principles of sensorimotor learning. *Nature Reviews Neuroscience*, 12(12):739, 2011.

D. M. Wolpert, Z. Ghahramani, and M. I. Jordan. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882, 1995.

D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

I. Yildiz and S. Kiebel. The cmu pronouncing dictionary. *http://www.speech.cs.cmu.edu/cgi-bin/cmudict*.

K. Yoshida, N. Saito, A. Iriki, and M. Isoda. Representation of others' action by neurons in monkey medial frontal cortex. *Current Biology*, 21(3):249–253, 2011.

T. R. Zentall. Imitation in animals: evidence, function, and mechanisms. *Cybernetics & Systems*, 32(1-2):53–96, 2001.

T. R. Zentall. Imitation by animals: How do they do it? *Current Directions in Psychological Science*, 12(3):91–95, 2003.

J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.