EPFL

Thesis n° 11175

Transferability of Learnt Speech Representations for Decoding Non-Human Vocal Communication

Presented on 8th August 2025

School of Engineering
L'IDIAP Laboratory
Doctoral program in Electrical Engineering

for the award of the degree of Docteur ès Sciences (PhD)

by

Eklavya SARKAR

Accepted on the jury's recommendation

Prof. V. Cevher, jury president
Dr J.-M. Odobez, Dr M. Magimai Doss, thesis directors
Dr M. Miron, examiner
Dr M. Cernak, examiner
Prof. D. Van De Ville, examiner

École
polytechnique
fédérale
de Lausanne

2025

To my late grandparents,
Dr. Ratna and Prof. Manoj Kumar Sarkar,
who dedicated their entire lives to science and teaching.

Acknowledgments

I would like to express my gratitude to all the people whose support has made this thesis possible. First and foremost, I thank my main supervisor and thesis co-director, Dr. Mathew Magimai-Doss, for giving me this valuable opportunity. I entered the field of speech processing with no prior experience or knowledge, yet leave with several published conference and journal papers. This was largely due to his broad knowledge, patience, willingness to teach, and especially his gentle nature, all of which have been key to my growth as a researcher. Whenever I faced an obstacle, no matter the topic or technicality, I could always turn to him for advice and leave with a number of ideas for surmounting it. I am sincerely grateful to have had a mentor who genuinely cares about the development and well-being of his students.

I also thank Dr. Jean-Marc Odobez for agreeing to be my thesis director. My gratitude extends to the jury members, namely Dr. Marius Miron, Dr. Milos Cernak, and Prof. Dimitri Van De Ville, as well as the jury president, Prof. Volkan Cehver. I am also indebted to Idiap's administrative and technical staff for their constant support, especially from Frank Fomaz, Louis-Marie, and Laura Coppey, who made me feel welcome at Idiap since my first day.

Most of all, I would like to express my gratitude to all the people at Idiap who felt like a true family and made the past five years an experience of a lifetime. I have never had a more international and tightly-knit group of friends, blending a wide range of cultures and backgrounds. In my twenty years in Switzerland, I have often felt a lack of openness towards other cultures, but my time here has filled that gap with a mountain of cherished memories, shared milestones, and personal growth. Idiap truly feels like a warm and welcoming home for people from all around the world, and I hope it always remains so.

To that end, I would like to first thank the people who helped me at the very beginning of my academic journey. I would not be here without the patient guidance of Laurent, my first Idiap friend, with boundless intelligence, and from whom I've perhaps learnt the most. I equally thank Parvaneh for supporting me in many times of need, as well as Apoorv for many engaging discussions, novel ideas, and being an overall inspiring role-model researcher and friend. My journey would have been undeniably harder without the presence of Zohreh to share my numerous anxieties and health struggles, and Amir, who helped solve many technical issues.

I also thank Fabio for his inclusivity, Roberto for his endless entertaining stories and thoughtful

conversations, Pablo for his constant willingness to help, Andrei for helping me move, François for his humour, Anshul for his friendship – at the office, gym, or on the dancefloor – as well as Arya, Pierre, Florian Piras, Karl, Haruki, Florian Mai, Tilak, Suhan, Sarthak, Enno, Julian, Evann, Michiel, Maxime, Carlos, Sergio, Mirko, Cem, and Andrea for their company. I am also very happy to have developed a close friendship with Darya and Barbara, both of whom play along with humour, hype me when needed, and ground me when necessary. I shared equally many lovely moments with Yulia, Louise, Chloe, Hande, Laura Vásquez, Imen, Sargam, Mingchi, Colombine, Vedrana, and Ina. Lastly, I would like to thank Neha for tirelessly celebrating everyone's birthdays and being the best cook anyone could ever ask for.

This thesis was funded by the Swiss National Science Foundation's NCCR Evolving Language project (grant no. 51NF40_180888), which provided the financial support for conducting this research. Over the course of this PhD, I traveled to five conferences, where I had the opportunity to present my work, connect with researchers in the field, gain valuable insights, and receive constructive feedback – all of which helped develop my self-confidence and gave me a sense of identity. I am grateful for the project's exceptional summer retreats and winter meetings, which allowed me to meet and befriend a whole new group of PhD students, postdocs, and PIs from other institutes working in linguistics, biology, philosophy, and neuroscience. The transdisciplinary nature of the project pushed me to expand my knowledge beyond machine learning and engineering, and into areas such as animal communication and the evolution of language, leading to successful collaborations. I feel exceedingly fortunate to have been part of such a vibrant Swiss initiative – one that made me feel part of a much larger collective and allowed me to experience many unique moments and milestones.

The idea of pursuing a PhD might never have occurred to me without the values instilled by both of my parents, Sharad and Satyajit, from a young age, especially to strive for excellence and to think beyond conventional paths. This PhD is as much their success as it is mine. I'm deeply thankful to them, as well as to my sister Aranya and my brother-in-law Marko, for their unwavering support throughout this journey. I especially thank my mother, who put me above other priorities and stayed with me in Martigny several times during the most critical periods of my PhD, taking care of my health and helping me meet paper and project deadlines on time, often under challenging conditions. I also thank Anant for always checking in on my well-being, being constantly available to listen to my issues, and providing emotional support.

Finally, I would also like to acknowledge my own journey – not just during this PhD, but everything that led up to it. I am proud to be submitting an EPFL thesis at long last, and to be defending it exactly a decade after dropping out from the same institute, as an undergraduate. It has been a long and often difficult journey, involving moving to different cities abroad and adapting to foreign cultures. And yet, my path eventually brought me back to where it had once nearly ended. Closing this chapter of my life, I now look forward to the next one, and will always look back on this one with fondness and a sense of accomplishment.

Martigny, July 23, 2025 Eklavya

Abstract

Humans and animals both use acoustic signals for vocal communication. The advent of self-supervised learning (SSL) has enabled neural networks to learn robust and general feature representations through the intrinsic acoustic structure of input signals, without prior knowledge or supervision. Given that both human speech and animal vocalizations are inherently structured signals that encode information, this thesis investigates whether representations learnt from human speech are transferable for decoding non-human animal vocalizations.

We first formulate and validate our core hypothesis through a proof-of-concept caller detection study on marmoset vocalizations, where multiple pre-trained SSL models are benchmarked. Building on this, we further evaluate their transferability across multiple marmoset datasets, and demonstrate that early layer representations from SSL models such as WavLM outperform traditional handcrafted features for call-type and caller identity classification.

We then explore how differences in auditory bandwidth between humans and animals influence the transferability of such SSL features. We show that bandwidth mismatches can have an impact on performance, and increasing its size yields a monotonic improvement for call-type and caller classification. We also compare SSL models pre-trained on speech with those pre-trained on general audio or directly on animal vocalizations. Our experiments reveal that general-purpose audio pre-training yields comparable performance to human speech pre-training, and the bioacoustics-trained models marginally improve it on specific datasets.

To further improve classification scores, we investigate model adaptation of the pre-trained SSL models. Fine-tuning such speech models on an automatic speech recognition task in a supervised framework does not bring any consistent improvements in performance, and in some cases, actually leads to a performance decline in the later layers. However, parameter-efficient fine-tuning strategies, such as Low-Rank Adaptation (LoRA), combined with selective layer freezing and pruning, achieves significant gains over standard linear probing in specific scenarios, while also reducing training complexity. Our results underscore the importance of LoRA adapter placements, layer selections, and fine-tuning strategies.

Finally, we attempt to leverage the sequential nature of animal vocalizations. While previous experiments temporally averaged extracted features into single vector representations, we use vector quantization frameworks to discretize frame-level SSL features into acoustic token

sequences. We evaluate these sequences through Levenshtein-distance analysis and sequence classification, and find that while they preserve some degree of acoustic discriminability, their performance remains well below that of a simple linear classifier applied to averaged functional vectors.

On the whole, this thesis demonstrates that SSL representations learnt from human speech can generalize effectively to animal vocalizations. Our work provides a practical and robust groundwork for computational bioacoustics, as well as a foundation for further bridging machine learning with animal communication science.

Keywords: bioacoustics, animal vocalizations, self-supervised learning, speech and audio feature representations, transfer learning, low-rank adaptation, vector quantization, bandwidth, call-type and caller classification, machine learning.

Résumé

Les humains et les animaux utilisent tous deux des signaux acoustiques pour communiquer vocalement. L'avènement de l'apprentissage auto-supervisé (AAS) a permis aux réseaux neuronaux d'apprendre des représentations de caractéristiques robustes et générales à partir de la structure acoustique intrinsèque des signaux d'entrée, sans connaissance préalable ni supervision. Étant donné que la parole humaine et les vocalisations animales sont toutes deux des signaux structurés qui véhiculent de l'information, cette thèse étudie si les représentations apprises à partir de la parole humaine peuvent être transférées pour décoder les vocalisations animales non-humaines.

Nous formulons et validons d'abord notre hypothèse principale à travers une étude de détection de l'appelant sur les vocalisations de ouistitis, en utilisant plusieurs modèles AAS pré-entraînés. En nous appuyant sur cette première analyse, nous évaluons ensuite leur transférabilité sur plusieurs ensembles de données de ouistitis, et montrons que les représentations des couches inférieures de modèles tels que WavLM surpassent les caractéristiques traditionnelles conçues manuellement pour les tâches de classification du type d'appel et de l'identité de l'appelant.

Nous explorons ensuite comment les différences de bande passante auditive entre humains et animaux influencent la transférabilité de ces représentations AAS. Nous montrons que les incompatibilités de bande passante peuvent affecter la performance, et qu'une bande passante plus large entraîne une amélioration monotone pour la classification du type d'appel et de l'appelant. Nous comparons également des modèles AAS pré-entraînés sur la parole humaine à ceux entraînés sur de l'audio général ou directement sur des vocalisations animales. Nos expériences montrent que les modèles pré-entraînés sur de l'audio général atteignent des performances comparables à ceux pré-entraînés sur la parole humaine, et que les modèles entraînés sur des données bioacoustiques peuvent légèrement les surpasser sur certaines bases de données.

Pour améliorer davantage les scores de classification, nous étudions l'adaptation des modèles AAS pré-entraînés. L'adaptation supervisée de modèles pré-entraînés sur la parole à une tâche de reconnaissance automatique de la parole n'apporte pas d'amélioration systématique des performances, et peut même entraîner une baisse dans les couches neuronales supérieures. En revanche, des stratégies d'adaptation efficaces en paramètres, telles que l'adaptation à

faible rang, combinées à un gel et une sélection partielle des couches neuronales, permettent d'obtenir des gains significatifs par rapport à un simple classificateur linéaire dans certains scénarios, tout en réduisant la complexité d'entraînement. Nos résultats soulignent l'importance du placement des adaptateurs, du choix des couches, et des stratégies d'adaptation.

Enfin, nous tentons de tirer parti de la nature séquentielle des vocalisations animales. Alors que les expériences précédentes moyennaient temporellement les caractéristiques extraites en un seul vecteur fonctionnel, nous utilisons des méthodes de quantification vectorielle pour discrétiser les représentations AAS en séquences de jetons acoustiques. Nous évaluons ces séquences à l'aide de l'analyse par distance de Levenshtein et de la classification de séquences, et constatons que, bien qu'elles conservent une certaine capacité de discrimination acoustique, leurs performances restent inférieures à celles d'un simple classifieur linéaire appliqué à des vecteurs moyens.

Dans l'ensemble, cette thèse montre que les représentations AAS apprises à partir de la parole humaine peuvent se généraliser efficacement aux vocalisations animales. Notre travail propose un cadre pratique et solide pour la bioacoustique computationnelle, et jette les bases d'un rapprochement entre apprentissage automatique et science de la communication animale.

Mots-clés : bioacoustique, vocalisations animales, apprentissage auto-supervisé, représentations de caractéristiques de la parole et de l'audio, apprentissage par transfert, adaptation à faible rang, quantification vectorielle, bande passante, classification du type d'appel et de l'appelant, apprentissage automatique.

Contents

List of Figures

List of Tables

List of Abbreviations

AB	AdaBoost
AI	Artificial Intelligence
AS	AudioSet
AUC	Area Under the Curve
BERT	Bidirectional Encoder Representations from Transformers
BPE	Byte-Pair Encoding
Catch22	CAnonical Time-series CHaracteristics
CLID	Caller Identification
CNN	Convolutional Neural Network
CTID	Call-Type Identification
DL	Deep Learning
DNN	Deep Neural Network
E2E	End-to-end
FC	Fully-Connected (Layer)
GeLU	Gaussian Error Linear Unit
GMM	Gaussian Mixture Model
GPT	Generative Pre-trained Transformer
GVQ	Gumbel-Softmax Vector Quantization
HCTSA	Highly Comparative Time-Series Analysis
HMM	Hidden Markov Model
IMV	InfantMarmosetsVox
***k*-NN**	k-Nearest Neighbours
LoRA	Low-Rank Adaptation
LVSM	Linear Support Vector Machine
ML	Machine Learning
MLP	Multi-Layer Perceptron
NCE	Noise Contrastive Estimation
NLP	Natural Language Processing
PEFT	Parameter-Efficient Fine-Tuning
RBM	Restricted Boltzmann Machines
ReLU	Rectified Linear Unit
RF	Random Forest

ROC Receiver Operating Characteristic
RW Raw-Waveform
SID Sex Identification
SL Supervised Learning
SSL Self-Supervised Learning
SVM Support Vector Machine
UAR Unweighted Average Recall
VQ Vector Quantization

1 Introduction

Bioacoustics is the study of animal sounds, specifically the production, transmission, and reception of acoustic signals in animals and their environments, and is often studied to understand the mechanisms underlying animal vocal communication (Bradbury and Vehrencamp, 1998). Animal vocalizations are of particular interest as they encode a range of critical information, spanning from individual and social behavior (Hauser, 1996; D. T. Blumstein et al., 2011) to species interactions, habitat health, and ecological dynamics. In addition, understanding bioacoustic signals can provide key insights into the foundational principles shared by human and animal communication systems (R. M. Seyfarth and D. L. Cheney, 2010; Fedurek, Slocombe, and Zuberbühler, 2016). Bioacoustics is thus also used to study the origins and evolution of language and vocal learning (Hurford, 2012; Fitch, 2018), as a means to deepen our understanding of communication in the non-human natural world.

Human vocal communication has been extensively studied and has progressed through successive stages of methodological innovation and refinement. Early speech processing systems relied on explicit models of speech production, most notably the source–filter model (Fant, 1970), as well as signal-processing theory. These foundations gave rise to methods such as linear predictive coding (Atal and Hanauer, 1971) and carefully engineered spectral features like Mel-frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980). However, the advent of artificial intelligence and deep learning (LeCun, Bengio, and Hinton, 2015; Schmidhuber, 2015) demonstrated that many of these hand-designed priors are no longer essential: rich, task-relevant representations can be learned directly from raw audio with just supervision and minimal domain knowledge (Hinton et al., 2012; Dahl et al., 2012; Graves, A.-r. Mohamed, and Hinton, 2013). More recently, self-supervised learning (SSL) has enabled models to learn robust, generalizable representations from the geometry of unlabeled speech data (Oord, Y. Li, and Vinyals, 2018), eliminating the need for direct supervision and annotated corpora. When combined with the availability of large-scale data, high-performance computing clusters, and novel transformed-based architectures (Vaswani et al., 2017), technologies such as automatic speech recognition (Baevski et al., 2020; W.-N. Hsu et al., 2021; Radford et al., 2023), speaker identification (Snyder et al., 2018b; Desplanques, Thienpondt, and Demuynck, 2020; Bai and

X.-L. Zhang, 2021), and text-to-speech synthesis (van den Oord et al., 2016; Shen et al., 2018) have progressed to unprecedented levels of performance.

By contrast, the study of non-human vocal communication, though rich in potential insights, still remains relatively underdeveloped, with comparatively little prior knowledge to guide research. Computational bioacoustics aims to 'decode' animal vocalizations to gain insights into their communication. In practice, this means automatically deriving information from animal signals through detection and classification tasks, such as vocalization detection, call-type classification, caller identification, sex classification. Early studies often relied on labour-intensive manual data annotation, and predominantly used spectograms as input representation from which further statistical features, such as peak frequencies, sound event durations, and more, were derived. Such studies typically only addressed small, species-specific datasets with a limited number of subjects, constraining the generalization and scalability across taxa and recording conditions. More general investigations often focused on broad tasks which are relatively easy, such as species classification, solvable using traditional machine learning classifiers. Moreover, a strong proportion of these studies were also exclusively focused on avian bioacoustics (Kahl et al., 2021; Ghani et al., 2023). Recent application of deep learning networks to bioacoustics has shown promise, enabling researchers to learn salient representations to analyze animal vocalizations at a larger scale (Stowell et al., 2019; Sainburg, Thielk, and Gentner, 2020). Notably, re-purposing deep learning architectures originally developed for human speech tasks for bioacoustics has shown some success (Y.-J. Zhang et al., 2018; E. Coffey et al., 2019; Bergler et al., 2019), suggesting potential domain transferability. Nonetheless, this deep learning approach remains a species-specific approach, and requires model training from scratch with supervision on large labeled datasets, which are still rare in bioacoustics.

Humans and animals both possess production and perception systems that allow them to communicate vocally through acoustic signals (Prather, 2013). In humans, speech is generated through a vocal production mechanism involving an excitation source, namely the vibration of the vocal folds, and the vocal tract system (Jurafsky and Martin, 2025). Similarly, animals also possess vocal production mechanisms (A. A. Ghazanfar and Rendall, 2008). Although the biological specifics may differ, the existence and use of such acoustic mechanisms is a shared commonality for vocal communication in humans and animals. The emergence of self-supervised learning, as a modern deep learning framework in speech and audio processing, has produced models capable of learning representations directly from the raw acoustic input, without incorporating any prior knowledge about the underlying production or perception systems. Instead, they learn by identifying intrinsic structure in the spectro-temporal patterns of the signal itself. Given that both human and animal vocalizations are inherently structured and non-random signals that encode meaning, this thesis investigates whether representations learnt from intelligible, high-resource human speech can transfer to the acoustic domain of animal vocalizations. We hypothesize that such representations, learnt in a self-supervised framework, can serve as a powerful prior for decoding complex, low-resourced animal vocal signals. Prior to this work, and to the best of our knowledge, no prior study had systematically explored this question. To investigate this hypothesis in depth, we formulate the following

central research questions (RQs) that we address in this thesis:

RQ1. Can representations learnt from human speech through SSLs be transferred to bioacoustic tasks, and if so, to what extent?

RQ2. How does a mismatch in auditory bandwidth between humans and the studied animal affect this transfer?

RQ3. Is this transferability specific to speech models, or can representations learnt from general audio also exhibit a similar cross-domain utility?

RQ4. Can adaptation of these pre-trained SSL models further improve the transferability ?

RQ5. How well can these transferred representations capture and leverage the sequential structure of animal vocalizations ?

By addressing these questions, this thesis aims to establish groundwork that can serve as a practical foundation for future computational bioacoustics studies.

1.1 Context and Motivation

This work is carried out within the NCCR Evolving Language, an interdisciplinary Swiss National Science Foundation initiative to explore the evolutionary origins and future of language and communication. As part of the Transversal Technology Task Force work package, the key motivation for this thesis is to help develop computational tools to support biologists, linguists, and ethologists in their research on human and animal communication. This line of research is especially useful for the following causes:

- **Conservation and biodiversity monitoring**: passive acoustic recording offers a non-invasive, scalable approach to track species presence, population, and behaviour over time. Automated analysis of these recordings can alert conservationists to habitat degradation, invasive species, or population decline without the need for costly field surveys. To that end, integrating robust bioacoustics representations into real-time sensor networks can facilitate continuous surveillance of remote habitats, and enable tracking of ecological disturbances, species migration patterns, and general animal welfare.
- **Comparative communication science**: animal vocalizations encode multiple layers of information, including individual identity, social intent, and environmental context, that follow the functions of human language. Decoding these signals with learnt representations can allow us to compare structural patterns, such as call repertoires or sequences, across species. By projecting vocalizations from diverse taxa into a common embedding space, researchers could investigate whether underlying semantic or

phonetic abstractions are shared, giving insight into the evolutionary pathways of vocal learning and information encoding.

1.2 Thesis Outline and Contributions

The structure of this thesis is axed around the defined research questions (RQ), and is organized as follows:

Chapter 2 provides the theoretical foundation necessary to investigate the aforementioned research questions. We review essential deep learning concepts and key speech and audio representations. Chapter 3 gives an overview of animal vocalizations and the type of information they encode, and presents the datasets employed in this thesis.

Chapters 4 and 5 both investigate RQ1. In Chapter 4, we formulate our core hypothesis on cross-domain feature transferability, and validate it with a proof-of-concept study on a caller identity detection task. We then extend this approach across multiple datasets and multi-class classification tasks in Chapter 5.

Chapter 6 addresses RQ2, where we examine the impact of the pre-training bandwidth on downstream bioacoustics classification tasks. Chapter 6 also investigates RQ3 by comparing performance of SSLs pre-trained on speech against those on general audio. Lastly, Chapter 7 completes the study by also examining SSLs pre-trained directly on animal vocalizations.

Chapter 7 and 8 both explore RQ4 in depth, studying various model adaptation strategies and fine-tuning domains for potential improvements in animal call classification performance.

Chapter 9 explores RQ5 by proposing feature representations based on discrete token sequences, and and evaluates them for animal calls. Finally, Chapter 10 concludes this thesis and suggests directions for future work.

Each chapter contains a schematic overview, publication note, and any supplementary material and collaboration notes. The schematic diagrams do not directly match the chapters' sections, but instead present a thematic overview.

2 Foundations of Deep Learning and Speech Representations

The goal of this thesis is to decode non-human bioacoustic signals by leveraging machine learning tools developed for high-resource human vocal communication. To that end, we employed a variety of machine learning and deep learning networks, concepts, and techniques. This chapter lays down the theoretical foundation for our work by first briefly reviewing essential deep learning concepts, architectures, and layers in Section 2.1. Building on this framework, we then explore various speech and audio representations relevant to bioacoustics, including traditional handcrafted features, representations learned through deep neural networks, and those derived from self-supervised learning and audio foundation models, in Sections 2.2 to 2.4 respectively. The chapter serves as a bridge between fundamental deep learning principles and the speech and audio methods used for extracting salient features from bioacoustic signals.

2.1 Deep Neural Networks

2.1.1 Deep Learning Framework

Given a large dataset $\mathcal{D}(\boldsymbol{x}, y)$ of paired input vectors $\boldsymbol{x}$ and target class labels y, deep learning aims to learn the underlying mapping from the inputs to the targets. In this framework, a deep neural network (DNN) approximates this mapping by modeling it as a parametric function f_θ, where the parameters θ are learned from the training data. For a given input $\boldsymbol{x}$, the network produces an output prediction $\hat{y} = f_\theta(\boldsymbol{x})$ intended to match the true target label y. In this context, the notion of *learning* or training refers to the process of finding optimal parameter values θ^* such that the network maps training inputs to their corresponding targets as accurately as possible.

The function f_θ is typically constructed as a composition of several simpler, differentiable sub-functions, commonly referred to as layers of the DNN:

$$f_\theta \coloneqq f^{(1)}_{\theta_1} \circ \ldots \circ f^{(L)}_{\theta_L}, \tag{2.1}$$

where L is the number of layers. Each layer l implements an affine transformation, and is characterized by its own set of parameters $\theta_l = \{w_l, b_l\}$, denoting the weights w and biases b. The term 'deep' in deep learning refers to networks with a high number of stacked layers.

The training process involves optimizing the parameters by minimizing a loss function $\mathcal{L}$ that quantifies the discrepancy between the network's predictions and the actual labels. This is typically achieved using gradient descent, an iterative method where the parameters are updated as follows:

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\theta_t), \tag{2.2}$$

where θ_t represents the value of the parameters of the model at iteration t during training, and η the learning rate. The gradients $\nabla \mathcal{L}(\theta_t)$ are efficiently computed using backpropagation, which applies the chain rule through the network layers. The overall training procedure is summarized in the following three main steps as depicted in Figure 2.1:

1. **Forward pass**: calculates the activations for each layer using the inputs $\boldsymbol{x}$ and the current parameters θ of the model, to predict an output $\hat{y}$.

2. **Backward pass**: computes the the gradients of the loss $\nabla \mathcal{L}$ with respect to the activations and parameters θ by propagating the error backwards through the network using the chain rule.

3. **Gradient step**: updates the existing parameters θ of the model using equation (2.2).

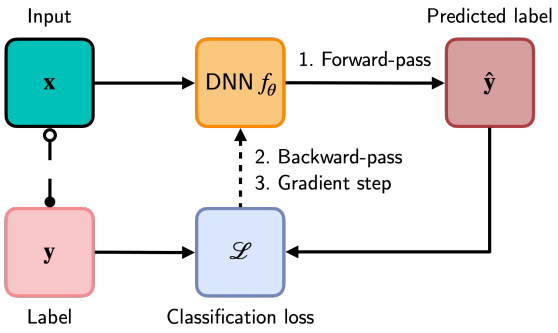


Figure 2.1 – The deep learning framework. The loss function measures the quality of the network's output and provides a feedback signal to adjust the model parameters.

Once training is complete, the learned parameters are frozen, and the model's generalization capability is typically evaluated on an unseen test set. Moreover, the network can also serve as a *feature extractor*, as the *embeddings* produced by its layers capture meaningful representations of the input data. In this thesis, we predominantly work with pre-trained models, analyzing the representations learned across different layers.

In short, a network transforms its input data into meaningful outputs, a process learned from exposure to data samples and their labels. At its core, deep learning is about meaningfully

transforming data, i.e. **learning useful representations** of the training dataset distribution that bring us closer to the desired targets. This representation learning can also be understood from a geometric perspective: the model applies a sequence of geometric transformations, with the aim of learning disentangled representations of continuous and complex data manifolds in high-dimension spaces, such that this space can be cleanly separable by class. Together with the growing availability of data, improvements in computational hardware, and algorithmic advances to the deep learning framework, this approach has driven the modern era of AI.

The following subsections provide a brief overview of the key deep learning architectures and layers used in this thesis.

2.1.2 Linear Layer and Perceptron

The perceptron model was one of the earliest neural network models to see practical use in machine learning (Rosenblatt, 1957). Its classification rule can is expressed as:

$$\mathbb{R}^{\mathbb{D}} \to \mathbb{R} \tag{2.3}$$

$$\boldsymbol{x} \mapsto \sigma\big(\mathbf{w}\cdot\mathbf{x}+b\big). \tag{2.4}$$

where the weights $\boldsymbol{w} \in \mathbb{R}^{\mathbb{D}}$ and bias $b \in \mathbb{R}$ are the learnable parameters of the model, and $\boldsymbol{x} \in \mathbb{R}^{\mathbb{D}}$ is an input vector. The perceptron essentially consists of a single *linear* layer, $(\boldsymbol{w}\cdot\boldsymbol{x}+b)$, which performs an affine transformation. It is also known as a *fully connected* layer, because every component of $\boldsymbol{x}$ is multiplied by a dedicated weight in $\boldsymbol{w}$, and then summed together with a bias term b. It is then followed by an activation function $\sigma : \mathbb{R} \to \mathbb{R}$. The non-linear activation is what enables the model to learn more complex decision boundaries.

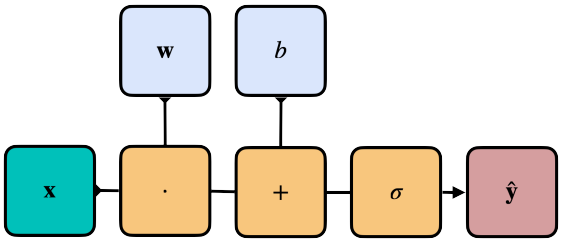


Figure 2.2 – Schematic representation of the operations in a perceptron model.

In recent years and in this thesis, a single linear layer (i.e., without an explicit non-linear activation function) is frequently employed as a simple classifier head on top of extracted feature embeddings from a frozen pre-trained network, particularly when the preceding layers have already learned a sufficiently rich representation of the data. Geometrically, the fully connected layer defines an afine transformation, whose zero-level set $\{\boldsymbol{x} \in \mathbb{R}^D : \boldsymbol{w}\cdot\boldsymbol{x}+b=0\}$, defines a decision boundary as a hyperplane which divides the input space into two separate regions. Often, the extracted embeddings are sufficiently linearly separable that this simple classifier can effectively distinguish between classes. As such, a fully connected layer not

only serves as a fundamental building block in many modern DNN architectures but is also commonly used at the end of a model to produce final class posterior probabilities.

In recent years and in this thesis, a single linear layer (i.e., without an explicit non-linear activation function) is frequently employed as a classifier head on top of feature embeddings extracted from a frozen pre-trained network, particularly when the preceding layers have already learned a sufficiently rich representation of the data. Geometrically, this fully connected layer implements an affine transformation whose zero-level set, $\{\boldsymbol{x} \in \mathbb{R}^D : \boldsymbol{w} \cdot \boldsymbol{x} + b = 0\}$, defines a decision boundary in the form of a hyperplane that divides the input space into two regions. Often, the extracted embeddings are sufficiently linearly separable that this simple classifier can effectively distinguish between classes. As such, the fully connected layer serves not only as a fundamental building block in many modern DNN architectures but also as an effective classifier on its own.

2.1.3 Multiple-Layer Perceptron

The linear perceptron model can be extended to a multi-dimension output by applying a similar transformation to every output, where $\boldsymbol{w} \in \mathbb{R}^{\mathbb{K} \times \mathbb{D}}$, $\boldsymbol{b} \in \mathbb{R}^{\mathbb{K}}$, and σ is applied component-wise. For $\forall\, l = 1, \ldots, L$, we define a multilayer perceptron (MLP) as:

$$\mathbb{R}^{\mathbb{D}} \rightarrow \mathbb{R}^{\mathbb{K}} \tag{2.5}$$

$$\boldsymbol{x}^{(l)} \mapsto \sigma\left(\mathbf{w}^{(l)} \cdot \mathbf{x}^{(l-1)} + \boldsymbol{b}^{(l)}\right). \tag{2.6}$$

where l is the layer index. The intermediate layers between the input and output are referred to as the hidden layers.

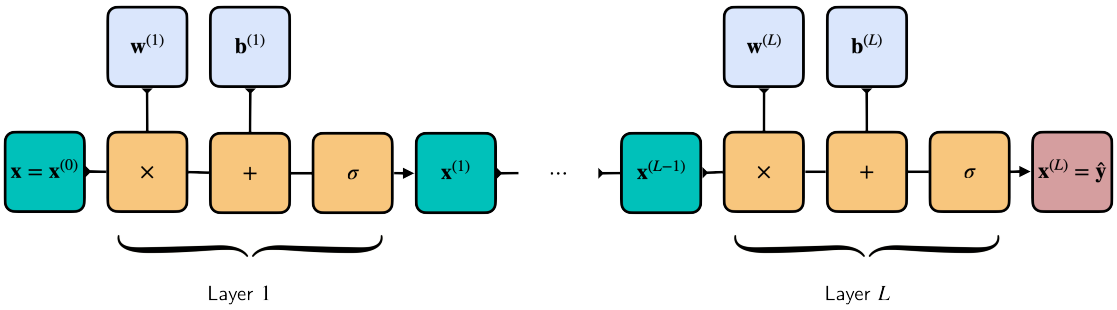


Figure 2.3 – Schematic representation of the operations in a multi-layer perceptron model

According to the universal approximation theorem, a single hidden-layer perceptron with sufficient neurons can approximate any continuous function on a compact domain (Hornik, Stinchcombe, and White, 1989). This theorem underscores the expressive power of even relatively simple MLP architectures, which is also often used a classifier head in this thesis.

2.1.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) (LeCun et al., 1989; LeCun and Bengio, 1998) extend linear layers by focusing on local patterns within the input. Instead of connecting every input unit to every output unit (as in a fully connected layer), a convolutional layer slides a small kernel or filter across the input, computing a weighted sum over each local region. This local connectivity allows the model to capture spatially or temporally localized features, making CNNs especially useful for speech and audio tasks.

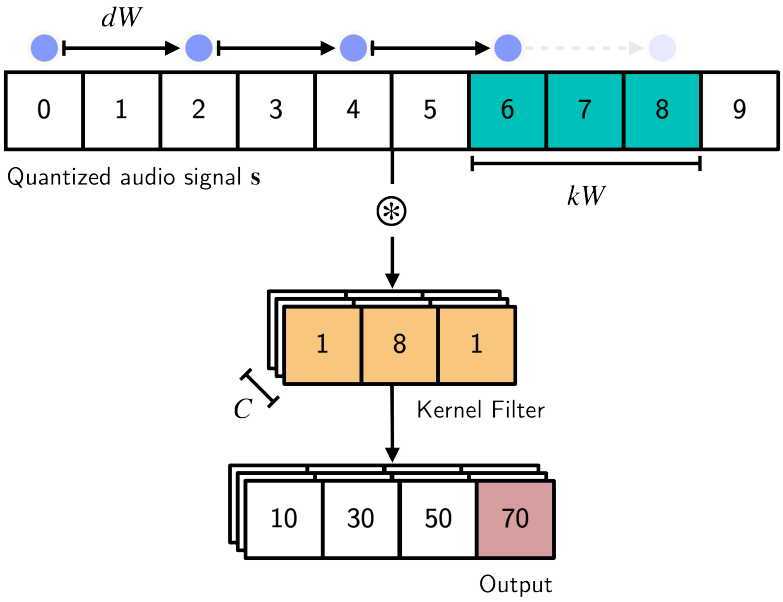


Figure 2.4 – 1D Convolutional layer applied to a signal $\boldsymbol{s}$. C represents the number of filters, kW the window length, and dW the window shift (•).

Figure 2.4 illustrates a one-dimensional convolution layer applied to an input signal $\boldsymbol{s}$. Each of the C filters is defined by a kernel of width kW, which is convolved with the signal in overlapping windows. Formally, for each position in the signal, the convolution output is obtained by element-wise multiplication of the filter weights and the corresponding segment of $\boldsymbol{s}$, followed by a sum:

$$\boldsymbol{x} \circledast \boldsymbol{kW} = \sum_{i=1}^{w} x_i \cdot kW_i, \tag{2.7}$$

Where the convolution's filter window length, also known as the kernel width, is denoted as kW. The stride dW, i.e. the window shift, determines how far the filter moves at each step. Because the operation is repeated locally, neighboring parts of the input influence neighboring parts of the output, preserving the signal's structure.

A convolutional layer is often paired with a max-pooling layer, which reduces the output dimension by taking the maximum value within a local window of length kW, shifted by dW, as presented in Figure 2.5).

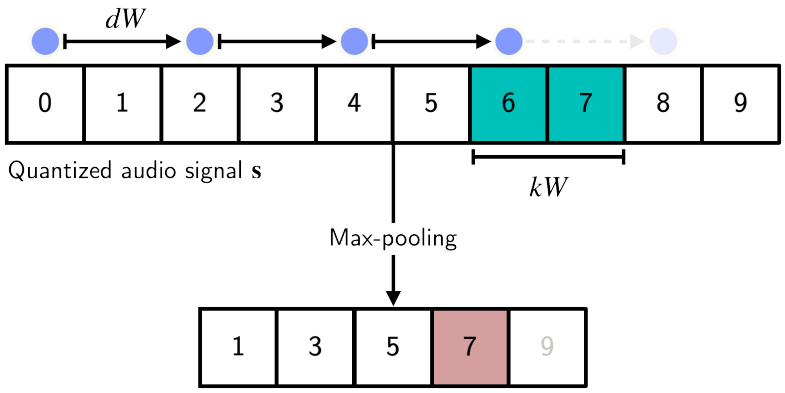


Figure 2.5 – Max-pooling applied to a signal $\boldsymbol{s}$. kW and dW are the window length and shift (•).

2.1.5 Attention and Transformers

Although networks based on fully-connected and convolutional layers have achieved considerable success, they also come with limitations. Fully-connected layers require fixed-size inputs and quickly become impractical for very high-dimensional data. Convolutional networks, while effective at capturing local patterns, need multiple layers to model long-range dependencies because each filter only covers a fixed-length context. In contrast, attention layers overcome these issues by capturing weighted interactions across all positions in an input sequence, making it easier to model long-range dependencies in high-dimensional or variable-length inputs.

In an self-attention block, the input sequence $\boldsymbol{x}$ is first linearly projected into queries Q, keys K, and values V. The attention weights $A \in \mathbb{R}^{N\times N}$ are then computed as:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right), \tag{2.8}$$

where D is the dimension of the keys K. The self-attention output of layer l is given by:

$$\text{SA}_l(x) = AV. \tag{2.9}$$

These weights determine how much each element in the sequence should contribute to the representation of every other element, effectively making the features context-aware.

A Transformer network is built by stacking multiple layers that combine self-attention with feed-forward blocks (implemented as one-hidden-layer MLPs), along with layer normalization and positional encoding. This architecture enables the model to capture global dependencies efficiently and is central to many modern pre-trained speech recognition models used in this thesis.

2.2 Handcrafted Speech and Audio Representations

Traditional approaches to speech and audio analysis rely on handcrafted features derived from expert knowledge in acoustics and signal processing. These knowledge-driven representations capture essential aspects of audio signals, such as frequency content, temporal dynamics, and spectral characteristics, that have long been instrumental in audio processing tasks. In this section, we outline several of these representations which have been used for bioacoustics in the literature, as well as in this thesis.

- Highly Comparable Time-Series Analysis (**HCTSA**) is an interpretable signal processing-based framework that has been demonstrated to be useful for diverse time series application domains (Fulcher, Little, and Jones, 2013). In this framework, a set of 7700 features are extracted by characterizing the signal by different time series analysis methods, such as, linear correlation, modeling fitting (e.g., autoregressive moving average analysis, GARCH), wavelet analysis, extraction of information theoretic measures, which then is combined with feature selection to build statistical models for the end task. In the literature, these features have been investigated for bioacoustics, namely behavioral birdsong discrimination (Paul et al., 2021), automated acoustic monitoring of ecosystems (Sethi, 2020), as well as marmoset caller identification (Phaniraj et al., 2023). One of the challenges of HCTSA approach is computational complexity and involves an evaluation of many similar features.
- In a recent work, CAnonical Time-series CHaracteristics (**Catch22**) features, a subset of the HCTSA feature set has been proposed which exhibit a strong performance across 93 real-world time-series classification problems, but are also minimally redundant (Lubba et al., 2019).

2.3 Deep Learning based Speech and Audio Representations

Based on the general concepts and networks outlined in Section 2.1, this section presents the speech and audio specific models developed with the advent of the deep learning framework. Unlike the knowledge-driven features in Section 2.2, the representations given below are learned automatically and purely from the audio data, without any specific assumptions.

- **End-to-end raw-waveform modeling** is a particular method in speech processing that leverages both end-to-end acoustic modeling and raw waveform modeling with a convolutional neural network. Figure 2.6 presents the complete pipeline for this network. The input audio signal $\boldsymbol{s}$ is send through multiple blocks of the 'feature learning stage', composed of a sequence of convolutional, max-pooling, and activation (typically TanH or ReLU) layers. Then, the embedding size is reduced by sending through an 1D adaptive average pooling, before flattening it and sending it through a final fully connected layer and obtaining the posterior probability distribution through the softmax activation

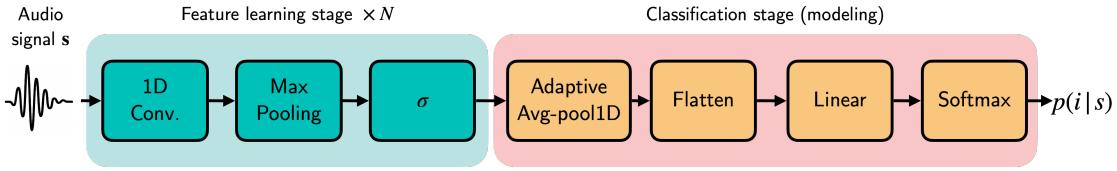


Figure 2.6 – Complete end-to-end raw-waveform pipeline. The input is the raw audio signal $\boldsymbol{s}$, and the output is the posterior probability distribution $p(i|x)$ for each class i. σ represents an activation function.

function. It is to be noted that the kernel filters in the convolutional layer are learned during training, and the first convolutional layer cab be seen as signal processing filters as they operate directly on the raw waveform (Palaz, Magimai.-Doss, and Collobert, 2019). To that end, the cumulative frequency response of these filters have been used to gain a deeper understanding and interpretability of the information that these end-to-end raw-waveform models learn during training (Muckenhirn, Magimai.-Doss, and Marcel, 2018; Muckenhirn et al., 2019).

- **Supervised features**: Another strategy is to leverage models pre-trained in a supervised fashion on large-scale audio datasets. One such example is the *Pretrained Audio Neural Network*, or **PANN** (Kong et al., 2020), specifically the `CNN14` architecture, which has been trained on AudioSet, a large corpus of diverse general audio recordings. In contrast to end-to-end raw waveform modeling, PANN operates on log-mel spectrogram inputs, learning both spectral and temporal patterns of sound events.

  In practice, `CNN14` processes the extracted spectrograms with six 2D convolutional blocks. Each block is composed of two convolutional layers with batch normalization and ReLU activations, followed by an average pooling operation that progressively reduces the time-frequency resolution while capturing increasingly abstract representations of audio events. Finally, a linear layer produces a 2048-dimensional embedding that we can extract and use as a general-purpose audio representation. This approach harnesses the strong generalization capabilities of supervised learning on extensive labeled data, enabling robust feature extraction for downstream tasks.

2.4 Self-Supervised Speech and Audio Representations

Self-supervised learning (SSL) offers an alternative approach to speech and audio representation, one that does not require prior knowledge or target labels of input data. Instead, SSL leverages vast amounts of unlabeled audio by training models to solve pre-text tasks, thereby learning rich, transferable representations. In contrast to the supervised features discussed earlier, which are learned from explicitly annotated datasets, SSL methods exploit the inherent structure of the data, allowing them to capture complex acoustic patterns that can be adapted to a wide range of downstream tasks. In the following sections, we outline the historical evolution of SSL in speech processing and detail a general framework consisting of pre-training on

unlabeled data followed by task-specific fine-tuning.

2.4.1 Historical Development

The emergence of self-supervised learning in speech processing can be understood through three distinct developmental stages:

1. **Clustering and mixture models**: Initial methods involved semi-automatic clustering of speech patterns using algorithms such as k-means, enabling recognition of isolated words by matching test samples to the nearest training clusters. Advances led to subword units being modeled using Gaussian Mixture Models (GMMs). Hidden Markov Models (HMMs) introduced dynamical modeling, supporting recognition of continuous speech rather than isolated words. These generative models (GMM/HMM) were typically trained by maximizing data likelihood, employing both supervised and unsupervised training strategies. Generative models were also utilized to extract informative speech features, leveraging their learned representations for downstream tasks such as speech recognition, speaker identification, and language verification.

2. **Stacked neural models**: The second wave transitioned from generative mixture models to neural network-based approaches, inspired by advances in representation learning techniques from computer vision and natural language processing (NLP). Compared to GMMs, neural architectures provided greater flexibility and capacity for modeling diverse input signals. Techniques such as restricted Boltzmann machines (RBM), denoising autoencoders, noise contrastive estimation (NCE), sparse coding, and energy-based models emerged, initially within vision and NLP contexts, before adaptation to speech tasks.

3. **Learning through pre-text tasks**: A more recent shift has been toward directly optimizing neural networks end-to-end using carefully designed pre-text tasks. Unlike earlier methods relying on layer-wise training, third-wave approaches involve training all network layers jointly. These methods frequently utilize very deep neural architectures, often exceeding ten layers, and evaluate learned representations on diverse benchmark tasks such as SUPERB for speech. The cornerstone of this third wave lies in pre-text task design, allowing effective use of knowledge from large unlabeled datasets. Popular tasks include generating complete information from partial inputs—such as predicting masked tokens (BERT series) or next tokens in sequences (ELMo, GPT)—and contrastive learning, where models learn representations by differentiating target instances from negative samples.

2.4.2 SSL Framework and Pre-Text Tasks

Figure 2.7 depicts the typical two-stage framework of self-supervised learning (A. Mohamed et al., 2022). It can be summarized as follows:

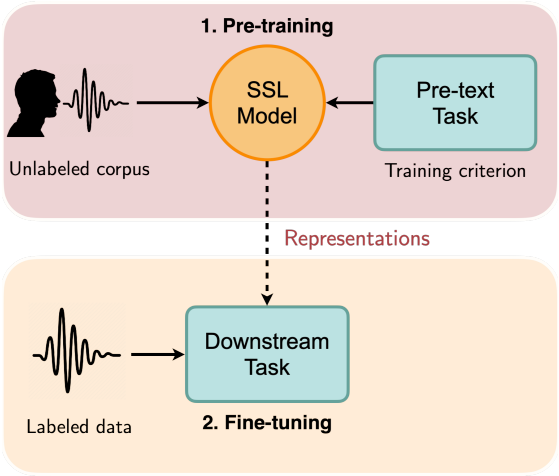


Figure 2.7 – Self-supervised learning two-stage framework.

1. **Pre-training**: The network is first pre-trained on vast amounts of unlabeled data using self-supervised objectives. During this phase, the model learns to extract meaningful and transferable representations by solving carefully designed pre-text tasks, such as predicting masked segments, reconstructing corrupted inputs, or contrasting similar and dissimilar samples. This process enables the network to capture rich, underlying structures in the data without relying on explicit labels.

2. **Fine-tuning**: Following pre-training, the learned representations are adapted to specific downstream tasks in the fine-tuning stage. Here, the pre-trained model is either further trained on a smaller labeled dataset or its fixed embeddings are used as input features for task-specific classifiers. Fine-tuning allows the network to tailor its generic, self-supervised features to the particular requirements of applications such as speech recognition, speaker identification, or other audio classification tasks.

The pre-text tasks for speech and audio SSL networks can be broadly categorized into the following four groups:

1. **Autoregressive reconstruction**: In this approach, the model is trained to generate future frames in an autoregressive framework. By learning to generate upcoming segments based on past context, the network implicitly captures the temporal dynamics and structure of the audio signal. Models such as APC (Chung et al., 2019) and VQ-APC (Chung, Tang, and Glass, 2020) both operate on spectrogram representations, and utilize this strategy, where the sequential prediction task forces the network to encode both local and global dependencies.

2. **Masked reconstruction**: This category involves reconstructing portions of the input signal that have been intentionally masked out. Unlike autoregressive methods that predict future frames, masked reconstruction tasks require the model to fill in missing

acoustic frames, encouraging it to learn contextual information from both preceding and following segments. Models such as NPC (A. H. Liu, Chung, and Glass, 2021), Mockingjay (A. T. Liu et al., 2020), and TERA (A. T. Liu, S.-W. Li, and Lee, 2021) employ this approach on a spectrogram basis. The approach is analogous to image inpainting in computer vision, and it benefits from the network's ability to model the overall structure of the audio spectrum.

3. **Masked prediction**: The network is trained to predict discrete pseudo-labels for the masked regions instead of directly reconstructing the raw acoustic features. This task forces the model to abstract the input signal into a higher-level, categorical representation, capturing salient characteristics that can be beneficial for downstream tasks. Models such as HuBERT (W.-N. Hsu et al., 2021) and WavLM (S. Chen et al., 2022) adopt this framework directly on the raw waveform. The learning process here bridges the gap between unsupervised feature extraction and supervised classification by encouraging the network to focus on the most informative parts of the input.

4. **Contrastive learning**: Contrastive approaches train the model to distinguish between similar (positive) and dissimilar (negative) samples. By formulating the learning problem as one of discriminating between correct and incorrect pairings, contrastive methods encourage the network to learn representations that cluster similar audio events together while pushing apart representations of different events. This method, employed by models such as Modified CPC (Riviere et al., 2020) and Wav2Vec2 (Baevski et al., 2020), operates directly on raw waveforms using convolutional layers. By framing the task as one of similarity learning, it leverages deep networks to capture both local and global contextual cues.

2.5 Bioacoustics Features

In very recent years, researchers have begun to pre-train models directly on bioacoustics data, marking a departure from earlier approaches that relied on the transferability of speech and general audio representations. While the previous sections described handcrafted features, deep learning models, and self-supervised techniques developed primarily on human speech or large-scale general audio datasets, direct pre-training on bioacoustics aims to capture species-specific acoustic patterns and other biological nuances.

One of the first and most comprehensive approaches in this domain is the AVES model family (Hagiwara, 2023a), which trains using HuBERT's architecture but on animal vocalizations instead of human speech. The AVES models are pre-trained using a masked-prediction task on a mixture of publicly available audio datasets—including FSD50K (Fonseca et al., 2021), AudioSet (Gemmeke et al., 2017), and VGGSound (H. Chen et al., 2020), thus exposing the model to a diverse range of bioacoustic signals.

2.6 Feature Extraction and Classifiers

Many of the networks described earlier are used in this thesis in their frozen, pre-trained form to leverage the robust representations they have already learned from large-scale data. Freezing these networks not only reduces computational and data requirements during our experiments, but also allows us to focus on evaluating the saliency of these representations, by training a separate classifier head without altering the underlying network or the extracted features. Figure 2.8 illustrates this pipeline: an input audio signal $\boldsymbol{s}$ is passed through the frozen feature extractor $\mathcal{F}$ to obtain a feature vector $\boldsymbol{x} \in \mathbb{R}^D$. It can be feature embeddings averaged on the temporal axis, or a single feature vector obtained as handcrafted representations. This vector then serves as input to a classifier head, with parameters θ, which is trained with backpropagation to predict the class label $\hat{y}$.

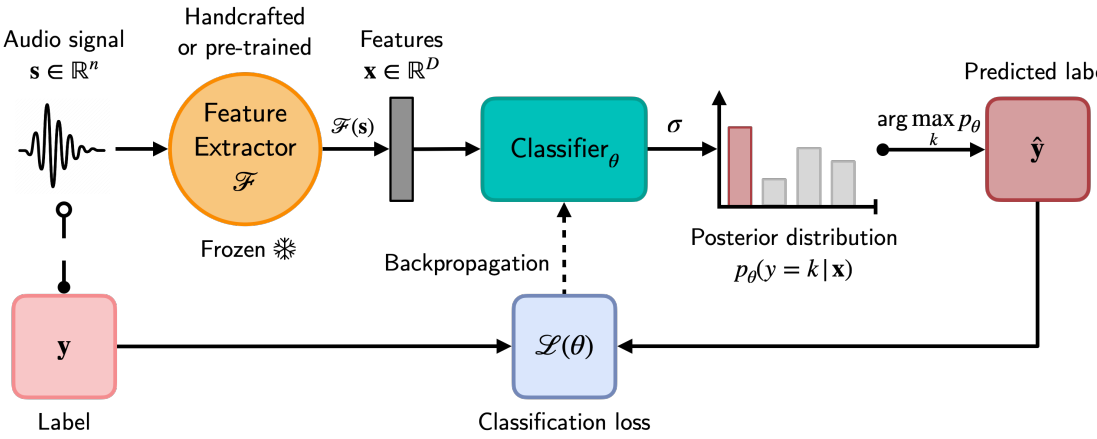


Figure 2.8 – Feature extraction and classification pipeline of a single layer.

By freezing the feature extractor, we retain the broad, domain-relevant information learned during pre-training, while training the classifier to the specific downstream task at hand.

In addition to using deep learning-based classifiers such as linear layers and MLPs (see Section 2.1), we also explore traditional machine learning classifiers. These methods are based on well-established statistical principles, operate independently of deep learning frameworks. Furthermore, unlike deep learning models, which typically require large amounts of data, these can perform effectively even on smaller datasets. The following traditional ML classifiers were used in our experiments:

- **Support Vector Machines (SVMs)** operate by first mapping input data into a high-dimensional feature space, in which the decision boundary can be represented as a hyperplane. Then, they identify the optimal hyperplane by maximizing the margin between the positive and negative classes. Initially developed as the *maximum margin classifier* (V. N. Vapnik and Lerner, 1963), the method evolved into *support vector classifier* or *soft-margin SVM* through the introduction of a soft margin (Cortes and

V. Vapnik, 1995). It further advanced to *support vector machines* by incorporating kernel methods (Boser, Guyon, and V. N. Vapnik, 1992), enabling non-linear boundaries, and subsequently generalized to multi-class classification framework (C.-W. Hsu and Lin, 2002).

- **Decision Trees**: Decision trees partition data by successively splitting it based on feature values, using measures such as Gini impurity or information gain, to arrive at a final decision at the leaf nodes. Building on this concept, **Random Forests** (RF) create an ensemble of decision trees by training each on random subsets of data and features, with the final prediction determined by aggregating the individual trees' votes (Breiman, 2001). **AdaBoost** (AB), in contrast, constructs a sequence of simple decision trees (often shallow ones known as decision stumps) where each subsequent tree focuses on correcting the errors made by its predecessors (Freund and Schapire, 1997). Together, these ensemble methods demonstrate how combining multiple models can lead to more robust and accurate predictions than any single decision tree alone.

2.7 Classification Evaluation Metrics

A classification model can either correctly classify a sample in its actual class, or it can incorrectly predict it to be in another class. When comparing the predicted class with the ground truth, we can obtain true positives (TP), true negatives (TN), as well as, false positives (FP) and false negatives (FN). Based on these, on can compute additional metrics, as given below:

- **Accuracy**: The proportion of correctly classified samples over the total number of samples. This metric provides a general measure of a model's overall performance.

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

- **Precision**: The ratio of true positive predictions to the total number of positive predictions made by the model. It reflects the model's ability to avoid false positives.

$$\text{P} = \frac{TP}{TP + FP} \tag{2.11}$$

- **Recall**: Also known as sensitivity or the true positive rate (TPR), recall is the ratio of true positive predictions to the total number of actual positive instances. It indicates how effectively the model identifies all relevant cases.

$$\text{R} = \frac{TP}{TP + FN} \tag{2.12}$$

- **F1**: The harmonic mean of precision and recall. The F1 score balances both metrics to

provide a single measure that accounts for both false positives and false negatives.

$$\text{F1} = \frac{2 \cdot P \cdot R}{P + R} \tag{2.13}$$

- **AUC**: The Area Under the Receiver Operating Characteristic (ROC) Curve. AUC measures the model's ability to distinguish between classes across all possible classification thresholds. It essentially gives a number to the ROC curves, which tell us the strength of classification rates in numbers. A ROC-AUC curve can be visualized by plotting a classifier's type TPR against its FPR, as shown in Figure 2.9a). We ideally want the ROC curve to be as close as possible to the ideal (0,1) point, and thus the AUC to be as close to 1 as possible.
- **Confusion Matrix**: allows one to visualize the accuracy of a model's classifier by comparing its predictions against the ground truths for each class. Figure 2.9b) shows what an ideal normalized confusion matrix would look like.
- **UAR**: Unweighted Average Recall is the mean recall calculated across all classes, treating each class equally. This metric is particularly useful in scenarios with imbalanced class distributions, and is therefore extensively used throughout this thesis.

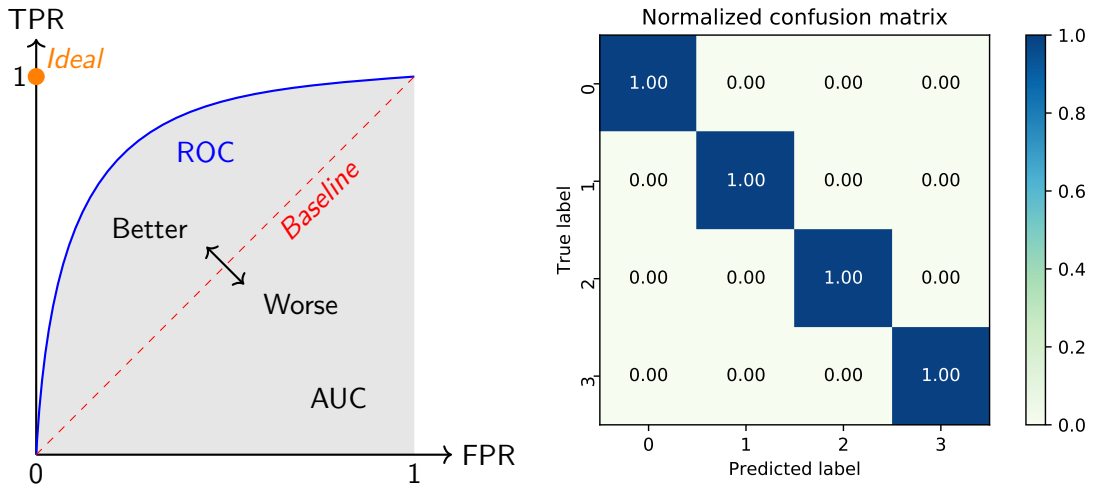


Figure 2.9 – **Left**: Sample ROC curve and its corresponding Area Under the Curve (AUC). The diagonal baseline represents a 'line of no-discrimination', and the (0,1) spot is the ideal classification point. **Right**: Ideal confusion matrix of 4 classes. The diagonal and off-diagonal cells respectively represent the model's normalized correct and incorrect class predictions rates.

2.8 Summary

This chapter provided an overview of the theoretical deep learning foundations as well as various speech and audio representations, which form the basis of this thesis. We began first by reviewing key deep learning concepts, including fundamental building blocks such as

linear layers and multilayer perceptrons, and progressed to more advanced architectures such as convolutional neural networks and Transformers. We then examined both handcrafted and learned representations of speech and audio signals, highlighting how deep learning and self-supervised approaches can automatically extract informative features without explicit knowledge-driven design. We also introduced a common pipeline for feature extraction, emphasizing how frozen pre-trained models can be leveraged for downstream tasks with minimal additional training. Lastly, we listed traditional machine learning methods to complement the neural network-based classifiers, and concluded with an overview of the evaluation metrics used to assess model performance. In the next chapter, we will take a deeper look at the actual animal vocalizations datasets and the bioacoustics tasks we aim to solve.

3 Animal Vocalizations

This chapter presents an in-depth overview of the types of animal vocalizations studied in this thesis and their associated bioacoustic classification tasks. Building on the theoretical foundations and representation learning techniques discussed in Chapter 2, we now focus on real-world bioacoustics data from non-human primates, marine mammals, and domestic dogs. These species provide acoustically diverse vocalizations that are well-suited for evaluating the transferability of speech-based representations across taxa. The tasks addressed include multi-class classification, such as call-type identification (CTID), caller identification (CLID), and, where applicable, sex classification (SID). Through these datasets, we aim to explore the unique acoustic properties of different animal vocalizations and demonstrate the potential of modern machine learning techniques for decoding animal vocal communication. We clarify that this thesis focuses only on vocalization-based animal communication, i.e. signals produced by a vocal tract, and does not investigate other communication modalities such as gestures or non-vocalization sounds.

Table 3.1 – Dataset descriptions and statistics. L denotes the total length [minutes], S the number of samples, n_{task} the number of classes, SR the sampling rate [kHz], μ the median length [ms].

Dataset	**Animal**	**S**	$\boldsymbol{L}$	**SR**	$\boldsymbol{n}_{\text{CTID}}$	$\boldsymbol{n}_{\text{CLID}}$	$\boldsymbol{n}_{\text{SID}}$	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$
IMV	Marmosets	72,920	464	44.1	11	10	–	127	375
Bosshard	Marmosets	13,808	37	300	7	8	2	117	181
Wierucka	Marmosets	4,901	138	125	12	8	2	1,037	1,687
Watkins	Mammals	1,697	295	–	32	–	–	1,701	71,245
Abzaliev	Dogs	8,034	137	48	14	80	2	655	1313

Table 3.1 presents a statistical summary of the used datasets. Section 3.1, 3.2, and section 3.3 provide an overview of the marmoset, marine mammal, and dog datasets, respectively, along with our motivation for studying them.

3.1 Marmosets

Marmosets are a central focus of this thesis, as their vocal behaviour provides a particularly valuable model for studying the evolutionary origins of human language. Their relevance to comparative communication science makes them especially well-suited for exploring how vocal signals encode socially and biologically meaningful information across species. Section 3.1.1 further motivates this focus and provides a detailed survey on marmoset call analysis.

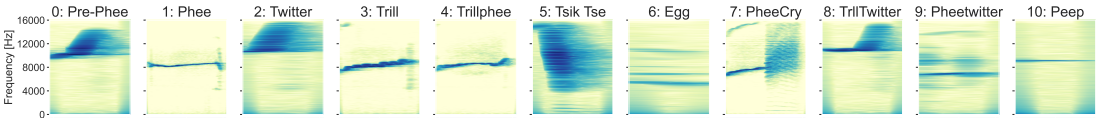


Figure 3.1 – Marmoset vocalizations by call-type.

3.1.1 Surrogate Models for Non-Human Primate Communication

Common marmosets (*Callithrix jacchus*) have recently gained prominence as a valuable research model among non-human primates. This is primarily due to their exceptional vocal abilities, which are rooted in their highly complex social behavior and cooperative breeding system (Eliades and Miller, 2017; Burkart et al., 2022). They possess extensive vocal repertoires used in various social situations (J. A. Agamaite et al., 2015; Bezerra and Souto, 2008), and their vocalizations have the capacity to encode a wide range of information, such as population, group affiliation, sex (Norcross and Newman, 1993), dialect (Zürcher and Burkart, 2017), and even individual caller identity (BS, DHR, and CK, 1993; Newman JD, 1992; Rukstalis and French, 2005; Phaniraj et al., 2023). These vocalizations are not limited to simple tonal signals but also encompass complex calls with multiple frequency components, some of which are within the ultrasonic range (J and JAM, 2018), and are expressed over a number of social and emotional states (Epple, 1968; R. Seyfarth and D. Cheney, 2003).

Moreover, marmosets have been observed to exhibit remarkable vocal adaptability. They can alter the duration (Brumm et al., 2004), intensity (Brumm et al., 2004; Eliades and X. Wang, 2012; Pomberger, Löschner, and Hage, 2020), complexity (Pomberger et al., 2018), or timing (Roy et al., 2011) of their calls, even when faced with disruptions in their environment that occur after the initiation of a call (Pomberger, Löschner, and Hage, 2020). These vocal characteristics align them closely with human speech properties, such as care-giving to infants, turn-taking (D. Takahashi, Fenley, and A. Ghazanfar, 2016), and categorical perception of sounds (Osmanski and X. Wang, 2023), and make them into a well-suited surrogate model for understanding the vocal communication of non-human primates among biologists (Worley and al., 2014) and neuroscientists (Okano, Miyawaki, and Kasai, 2015).

While these properties make marmosets an intriguing subject for the study of communication processes, they also pose a significant challenge when attempting to automate the analysis of their vocalizations. In the literature, the automatic analysis of marmoset vocalizations,

i.e. such as call-type, caller identity, or sex classification, has been conducted by leveraging signal processing features alongside traditional machine learning classifiers. (Turesson et al., 2016) compared different classification methods for marmoset call-type classification using linear prediction coefficients as feature representation, and found that on a small data setup of 30 samples per call-type, k-NN, SVM, and optimal path forest algorithms yielded better performance than multilayer perceptron, Adaboost, and logistic regression. (Wisler et al., 2016) investigated different feature representations, namely, audio features (statistics based on energy entropy, signal energy, zero crossing rate, spectral rolloff, spectral centroid, and spectral flux), mel-frequency cepstral coefficients (MFCCs), and Teager energy operator-based features for marmoset vocalization and call-type detection. On a synthetic dataset created by taking a small set of calls and augmenting it with background noise and acoustic events, it was found that feature-level combination led to better performance.

(Verma et al., 2017) investigated discovering different patterns in marmoset calls through unsupervised learning. Specifically, they developed an HMM-based approach to segment and cluster marmoset vocalizations into discrete units through multi-resolution and multi-rate analysis of the signal. In (Y. Zhang et al., 2018), it was demonstrated that marmoset vocalizations and call-types can be better detected and classified by feeding statistics of log-mel filter bank energies as input to recurrent neural networks (RNNs), when compared to SVM or multilayer perceptrons. In the scenario of analyzing recordings obtained from a pair of marmosets, (Oikarinen et al., 2018) investigated a deep learning approach where a spectrogram was fed as input to a convolutional neural network to jointly perform vocalization detection, call-type classification, and caller detection. It was found that joint modeling yielded better performance than training systems individually for each task in this scenario. Highly Comparable Time-Series Analysis (HCTSA) features have also been used to model source (caller) identification through an Adaboost-based hierarchical approach for marmosets (Phaniraj et al., 2023), as well as for 14 mammalian species (Wierucka et al., 2024).

Recent studies have begun exploring the self-supervised learning (SSL) framework, which leverages unlabeled data by creating surrogate labels from the data's inherent structure. This has led to works investigating birdsong detection (Saeed, Grangier, and Zeghidour, 2021a) and bioacoustic event detection (Bermant, Brickson, and Titus, 2022a) through contrastive pre-training. However, systematic investigations of self-supervised learning for animal vocal communication remain largely limited. In particular, their potential transfer from human speech to marmoset vocalizations holds great promise for uncovering cross-species representational similarities that may shed light on the evolutionary origins of language.

3.1.2 Datasets

- **InfantMarmosetsVox (IMV)** (Sarkar and Magimai.-Doss, 2023) is an extended version of the dataset used in the study on marmoset call type discrimination by (Y. Zhang et al., 2018). The dataset consists of 72,920 audio segments representing 11 different

call-types, and amounting to 464 minutes of vocalizations. The data contains 350 files of precisely labeled 10-minute audio recordings across all ten caller classes. The audio was recorded from five pairs of infant marmoset twins, each recorded individually in sound-proofed rooms at 44.1 kHz SR, without communication with other marmoset pairs or the experimenters. The audio recordings were manually labeled by an experienced researcher using the 'Praat' tool. For each vocalization, the start and end time, call type, and marmoset identity are been provided. Although a large dataset by bioacoustics standards, each segment is predominantly short, at a median length of 127 ms. The spectral range of the calls is mostly centered at around 7-8 kHz, although there is still some information present above 16 kHz (Sarkar and Magimai.-Doss, 2024). The calltypes are entitled peep (pre-phee), phee, twitter, trill, trillphee, tsik tse, egg, pheecry (cry), trllTwitter, pheetwitter, and peep calls.

- The **Bosshard** (Bosshard, 2020; Bosshard et al., 2024) dataset consists of 102 labeled 10-min focal audio recordings of common marmoset calls recorded in six behavioural contexts. A pair of marmosets was either separated or in the same enclosure, with preferred food either freely available for the focal individual or not. Each of the 8 subjects was recorded on 16 separate occasions. Most of the calls were given in bouts as holistic single call units, and thus, a call-type unit was defined as a call bout with call elements which were not further apart than 0.5s, as per existing literature (J. A. Agamaite et al., 2015; Snowdon and Elowson, 2001). We only used the segments labeled as single call elements, i.e. not split up in bouts, to avoid data overlap and duplication. The dataset consists of 7 calls, namely alarm, ek, food, phee, trill, tsk, and twitter. The audio recordings were manually annotated by using Avisoft SASLab Pro (Avisoft Bioacoustics, Feb. 2017) to narrowly label the start and end of each call-type. The data was collected under Swiss legislation and licensed by Zurich's cantonal veterinary office (license ZH 223/16 and ZH 232/19).

- The **Wierucka** dataset was collected from 6 target adult common marmosets, 3 male and 3 female, housed at the University of Zurich. Two additional non-target individuals were also included in the dataset, summing to 8 individuals in total. The data consists of 12 calls classes: phee, trill, food call, tsk, low tsk (tsk with a peak frequency of approximately 7-9 kHz), twitter (sequence), ek, phee sequence (multiple phees), low tsk sequence (multiple low tsks), ek sequence (multiple eks), food call sequence (multiple food calls). All procedures were done in accordance with Swiss legislation and were licensed by Zurich's cantonal veterinary office (license ZH223/19). For each recording, two individuals (one male and one female) were placed in adjacent wire cages and recorded simultaneously in 15-minute intervals with two UltraSoundGate 116H recorders coupled with an Avisoft CM16/CMPA condenser microphone (Avisoft Bioacoustics, Germany), each set to a different gain to capture both low and high amplitude calls with a sampling rate of 125kHz. A total of 12 recordings, spread over 7 months, were made for each target individual. Caller identity was labeled in real time using Avisoft-RECORDER USGH (Avisoft Bioacoustics, Germany). The labelling of the calls' exact start and end points was carried out

through a visual examination of the spectrograms. For inclusion in subsequent analyses, calls needed be distinctly visible on the spectrogram, devoid of any interference from other calls, and readily classifiable into specific call-type categories.

3.2 Marine Mammals

Marine mammal vocalizations are characterized by a wide range of acoustic features due to the diverse species and their varied communication contexts. These vocalizations often exhibit significant variation in frequency content and temporal structure, reflecting the adaptations of these animals to their underwater environments.

The **Watkins** dataset (Sayigh et al., 2017) contains the recordings of different marine mammals, such as specific dolphins, whales, and seals. We chose Watkins for its multi-species vocalizations, rich acoustic variety, and high variance in segment lengths (figure 5.1). It has been commonly used for bioacoustic benchmarking, particularly for evaluating modern deep learning models (Hagiwara, 2023a; Hagiwara et al., 2023b). We chose the 'best of' cut of the original dataset, a selected subset from the original 15,000 samples in total, deemed to be of higher sound quality and to contain less noise. The final dataset contains 1697 vocalization segments from 32 different species, totalling to 295 minutes, with a median length of 1701s. The sampling rate (SR) varies according to the recorded species.

3.3 Dogs

Dog vocalizations offer another intriguing domain for bioacoustic research, where subtle differences in bark types and other sounds can convey distinct emotional states or intentions. In our study, we focus on datasets that capture a range of canine vocal behaviors—from aggressive or fearful barks to those associated with excitement or owner interaction.

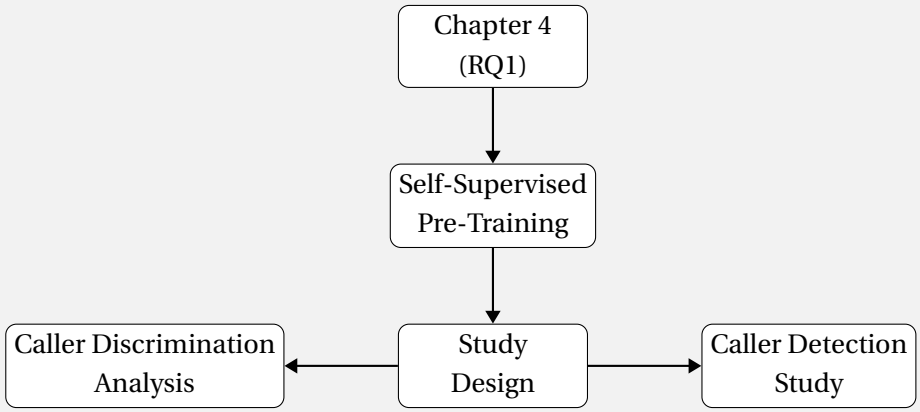
Abzaliev dataset is novel dog dataset (here referred to by the first author's name) consisting of 8,034 total vocalizations (Abzaliev, Perez-Espinosa, and Mihalcea, 2024). It contains 14 different call-types, ranging from normal, aggressive, fearful, and playful barks at strangers (IDs 0–3), to vocalizations related to owner interaction (4–5) and non-stranger/non-play sounds (6). It also contains postive or negative whines (7–8) and growls (9–10), barks associated with sadness or anxiety (11), and excitement upon the owner's arrival home (12). The recordings originate from various dog breeds, including Chihuahuas, French Poodles, and Schnauzers. The data was recorded at 48 kHz SR from a microphone, and followed a protocol designed and validated by experts in animal behavior. The dog vocalizations were induced by exposing the dogs to different types of external stimuli, with the participation of the owner and/or experimenter. We discard all the segments labeled as non-dog sounds, such as TV, cars, and appliances.

3.4 Summary

Together, these animal datasets provide the foundation for investigating how self-supervised representations learnt from human speech can be transferred to decode non-human vocal communication. In the following chapter, we begin this investigation with a proof-of-concept study on caller identity detection in marmosets, evaluating how well different SSL models can encode individual animal identity information from their vocalizations.

4 Proof of Concept: Leveraging SSL Representations for Caller Identity Detection

Chapter Schematic Overview



Publication Note

The material presented in this section is adapted from the following works:

- E. Sarkar and M. Magimai.-Doss (2023). "Can Self-Supervised Neural Representations Pre-Trained on Human Speech distinguish Animal Callers?" In: *Proc. of Interspeech*, pp. 1189–1193.

Supplementary Material

- **Source Code**: https://github.com/idiap/ssl-caller-detection.

4.1 Introduction

The study of animal vocalizations, or bioacoustics, has progressed significantly in recent years due to approaches inherited from machine learning and deep learning (Stowell, 2022a). However, most of these are supervised approaches, which require large amounts of labeled data, which is often scarce in bioacoustics. Self-supervised representation learning (SSL) has

emerged as a powerful tool in speech processing to leverage unlabeled data by pre-training models to solve pretext tasks using surrogate labels created from the structure inherent to the data itself. Given an acoustic waveform signal as input, an SSL model uses said labels and the pretext task to train and iteratively optimize its learning objective. The information encoded in the representations can vary depending on the selected learning objective, which can be roughly categorized into generative and discriminative approaches. Generative methods try to either reconstruct masked acoustic frames (A. H. Liu, Chung, and Glass, 2021; A. T. Liu et al., 2020; A. T. Liu, S.-W. Li, and Lee, 2021), or predict future frames using an auto-regressive framework (Chung et al., 2019; Chung, Tang, and Glass, 2020). Discriminative approaches either learn by contrastive learning, i.e. discriminating positive samples from negative ones (Riviere et al., 2020; Baevski et al., 2020), or else by predicting pseudo-labels of discrete masked regions (W.-N. Hsu et al., 2021; S. Chen et al., 2022; Baevski et al., 2022) or the output of specific hidden layers (H.-J. Chang, S.-w. Yang, and Lee, 2022). The representations learnt from the chosen SSL model can then be further fine-tuned to a wide range of speech downstream tasks, which have yielded state-of-the-art results on the SUPERB benchmark (S.-w. Yang et al., 2021).

Self-supervised learning only utilizes the intrinsic structure of unlabeled data without any reliance on domain-specific knowledge, such as human speech production, to capture essential information about the input data, and extract high-level representations in an embedding space. Thus, the utility of such representations may not only be restricted for modeling human speech, as demonstrated by recent works on other acoustic domains such as music (Wu et al., 2021; Zeng et al., 2021) and biomedical signals (Banville et al., 2021; Banville et al., 2019). Given this understanding, and the fact that both humans and animals have a voice production system, our objective is to investigate the cross-transferability of representations learned from human speech for analyzing animal vocalizations.

To that end, we conduct an animal caller detection study on Marmoset (*Callithrix jacchus*) vocalizations, and demonstrate its applicability through means of eleven different SSL models pre-trained with different pretext tasks. Our study also aims to provide practical benefits to biologists and ethologists by providing a framework to distinguish individual identities *within* the same animal species, which is an understudied topic in bioacoustics and a much harder problem than across-species classification (Stowell, 2022a). Some previous works has explored birdsong detection (Saeed, Grangier, and Zeghidour, 2021b) and bioacoustic event detection (Bermant, Brickson, and Titus, 2022b) using contrastive learning, however, the generalization of SSL models to animal vocalizations has largely remained unexplored. To the best of our knowledge, no previous study has looked into caller detection by utilizing the embedding space learnt by pre-training on human speech.

4.2 Study Design

This section presents the study design to systematically investigate the cross-transferability of representations learned from human speech for animal caller detection. Specifically, we

design a study with the following research questions:

1. How discriminative are the embedding spaces of SSL models pre-trained on human speech?
2. Can we systematically detect individual Marmoset callers using said embedding space?

The remainder of the section presents the dataset, research framework, and selection of SSL models for our investigations.

4.2.1 Dataset

For our study, we requested and used the marmoset dataset collected and labeled by (Y.-J. Zhang et al., 2018), defined as InfantMarmosetsVox (IMV) in Chapter 3. It contains audio recordings of eleven different marmoset calltypes, such as Twitters, Phees, and Trills, manually annotated using the Praat tool. The audio was recorded from five pairs of infant marmoset twins, each recorded individually in two separate sound-proofed recording rooms at a sampling rate of 44.1 kHz. The start and end time, call type, and marmoset identity of each vocalization are provided, labeled by an experienced researcher. The data contains 350 files of precisely labeled 10-minute audio recordings across all caller classes. We downsample the data to 16 kHz, remove all segments labeled as 'silence' and 'noise', and only keep the vocalization segments, amounting to a total of 464 minutes over 72,921 vocalization segments, with a mean and median length of 381 ± 375 ms and 127 ms respectively. Figure 4.1 shows the imbalanced distribution of vocalizations per caller, color coded by calltype. We divide the entire data into training, validation, and test sets, named *Train*, *Val*, and *Test* respectively, following a 70:20:10 split. This distribution allows us to train models on a sufficiently large dataset while ensuring that we have sufficient data for model evaluation and validation. *Train* is used to train the models, *Val* to tune hyperparameters, and *Test* to evaluate the trained models on unseen data.

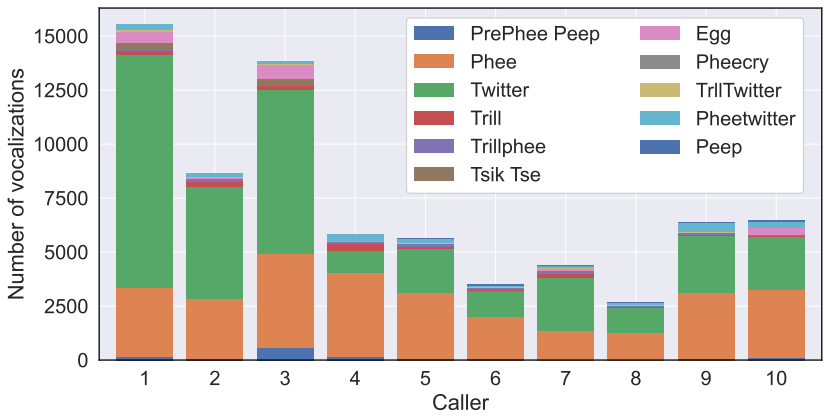


Figure 4.1 – Vocalization per callers grouped by call-type.

4.2.2 Caller-Groups

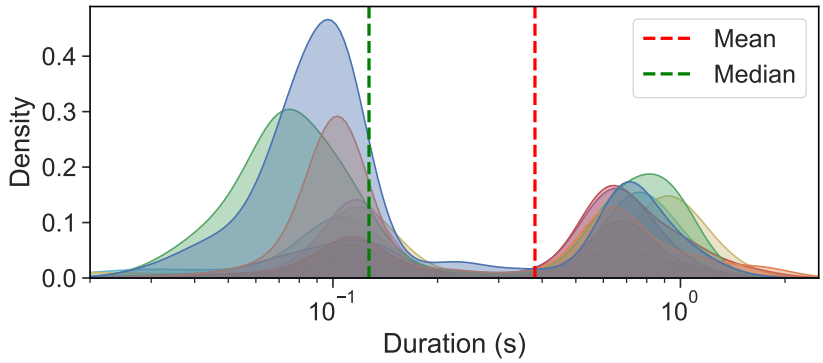


Figure 4.2 – Log distribution of vocalization lengths for callers 1–10 represented in different colors. The mean and median are calculated over the entire dataset.

For our study, neural embeddings are extracted from the pre-trained SSL models by giving the Marmoset vocalizations as input for the purpose of caller detection. The log distribution of vocalization lengths in this dataset, depicted in Figure 4.2, exhibits a bimodal structure consistent with prior findings (Huang et al., 2022; D. Y. Takahashi, Narayanan, and A. A. Ghazanfar, 2013). However, the same figure also illustrates that the vocalization segments in this dataset are predominantly short, with a median segment length of around 125 ms. Considering the lack of prior knowledge for this task, we took inspiration from i-vector and x-vector based speaker verification systems, where utterance lengths considerably longer than a short-term window size are modeled to achieve high performance (Dehak et al., 2011; Snyder et al., 2018a). More precisely, in order to effectively model each caller while accounting for the low vocalization segment length as well as to explore the acoustic variations within each caller, we first split all the vocalization embeddings by caller. Then, in order to maintain the chosen 70:20:10 split ratio of our data sets, we divide the embeddings of each caller sequentially into a fixed number of groups, hereafter referred to as 'caller-groups'. We set the number of said groups to 100 for *Train*, and proportionally scale for *Val* and *Test*. This results in a total of 1000, 280, and 140 groups across all callers for *Train*, *Val*, and *Test* sets, respectively.

4.2.3 Embedding Spaces

We carry out caller discrimination analysis and caller detection studies by computing the first and second order statistics of the SSL embeddings in the caller-groups. For this purpose, we select eleven pre-trained SSL models from the SUBERB leaderboard (S.-w. Yang et al., 2021) based on the different pretext tasks seen in Section 5.1, and use the S3PRL toolkit (S.-w. Yang et al., 2021) to extract the embeddings. Table 4.1 lists the chosen models, along with their number of parameters P in millions, and the dimension D of the last layer embedding. All the models have been pre-trained on the LibriSpeech (LS) corpus, except Modified-CPC which is pre-trained on the Libri-Light (LL) corpus.

Table 4.1 – Selected pre-trained SSL models on human speech. *P* indicates the number of parameters in millions, and *D* corresponds to the dimension of the last layer embedding.

Model	Corpus	*P*	*D*	Pretext Obj.
APC (Chung et al., 2019)	LS 360	4.11	512	Autoreg. Rec.
VQ-APC (Chung, Tang, and Glass, 2020)	LS 360	4.63	512	Autoreg. Rec.
NPC (A. H. Liu, Chung, and Glass, 2021)	LS 360	19.38	512	Masked Rec.
Mockingjay (A. T. Liu et al., 2020)	LS 100	21.33	768	Masked Rec.
TERA (A. T. Liu, S.-W. Li, and Lee, 2021)	LS 100	21.33	768	Masked Rec.
Mod-CPC (Riviere et al., 2020)	LL 60k	1.84	256	Contrastive
Wav2Vec2 (Baevski et al., 2020)	LS 960	95.04	768	Contrastive
Hubert (W.-N. Hsu et al., 2021)	LS 960	94.68	768	Masked Pred.
DistilHubert (H.-J. Chang, S.-w. Yang, and Lee, 2022)	LS 960	27.03	768	Masked Pred.
WavLM (S. Chen et al., 2022)	LS 960	94.38	768	Masked Pred.
Data2Vec (Baevski et al., 2022)	LS 960	93.16	768	Masked Pred.

4.3 Caller Discrimination Analysis

This section presents a discrimination analysis of SSL embedding spaces for the purpose of marmoset caller distinction. For this study we only use the *Train* portion of the data.

In order to conduct this analysis on our data, we first model the embedding spaces of each caller-group with a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu}$ and diagonal covariance matrix $\boldsymbol{\Sigma}$, resulting in a total of 100 multivariate Gaussians for each caller.

Subsequently, we compute the inter-caller and intra-caller distances by comparing the multivariate Gaussian distributions, as illustrated in Figure 4.3. Specifically, for inter-caller distances, we calculate a total of $100 \cdot 100$ pairwise distances for each pair of callers. For intra-caller distances, we compute a total of $\binom{100}{2}$ distances. To compute the distance between the the Gaussians of a pair of caller-groups, we use two measures, namely the Kullback-Leibler (KL) divergence and Bhattacharyya distance, both of which produce distances in the range of $[0, +\infty)$. The latter provides a symmetric measure while the former does not.

Equations 4.1 and 4.2 respectively provide the formulas for calculating the KL divergence D_{KL} and Bhattacharyya distances D_{BC} between two multivariate Gaussian distributions $\mathcal{N}_f$ and $\mathcal{N}_g$ (Durrieu, Thiran, and Kelly, 2012; Bhattacharyya, 1943). In the case of the KL divergence, the mean vector $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$, determinant $|\Sigma|$, and dimensionality d are utilized. Meanwhile, the Bhattacharyya distance uses the arithmetic mean of the covariance matrices $\boldsymbol{\Sigma_f}$ and $\boldsymbol{\Sigma_g}$ as $\boldsymbol{\Sigma}$.

$$D_{\mathrm{KL}}(f||g) = \frac{1}{2}\left(\log\frac{|\boldsymbol{\Sigma_g}|}{|\boldsymbol{\Sigma_f}|} + \mathrm{Tr}(\boldsymbol{\Sigma_g}^{-1}\boldsymbol{\Sigma_f}) + (\boldsymbol{\mu}_f - \boldsymbol{\mu}_g)^T\boldsymbol{\Sigma_g}^{-1}(\boldsymbol{\mu}_f - \boldsymbol{\mu}_g) - d\right) \tag{4.1}$$

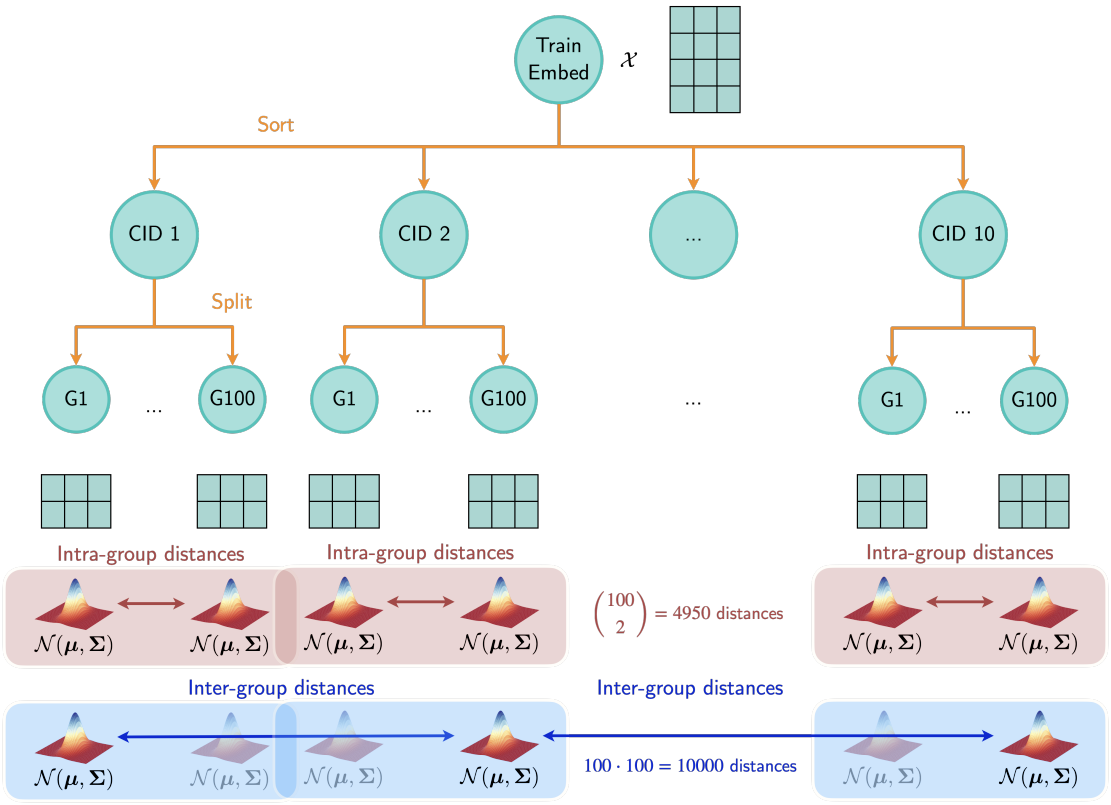


Figure 4.3 – We sort the *Train* embeddings by caller identity (CID1–10), and then split each of those into caller-groups (G1–100). We then model each caller-group's embedding spaces of with a multi-variate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and calculate the intra and inter-group distances.

$$D_{BC}(f||g) = \frac{1}{8}(\boldsymbol{\mu}_f - \boldsymbol{\mu}_g)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_f - \boldsymbol{\mu}_g) + \frac{1}{2}\log\left(\frac{|\Sigma|}{\sqrt{|\Sigma_f||\Sigma_g|}}\right) \tag{4.2}$$

Once we have computed the distribution of distances for all the SSL embedding spaces, we can visualize them through a heatmap. Figure 4.4 shows the distance matrix for WavLM's embedding space, where the diagonal entries represent the intra-caller distances and the off-diagonal correspond to the inter-caller distances. In an ideal scenario, one would expect the intra-class distances between distributions to be smaller than the inter-class ones, which is not entirely the case in our results. Nevertheless, for callers with a larger amount of available data, we can observe good discrimination when compared to callers with a lower amount of data, as in the case of Caller 1 and Caller 3 vs. Caller 8. We observe that the distances exhibit similar patterns for all other SSL embeddings, which suggests these embeddings provide similar information for the caller discrimination task. Taken together, the analysis suggests that the SSL embeddings do carry information for distinguishing marmoset callers to a certain extent. However, accomplishing this simple with a linear classifier may be a challenging task.

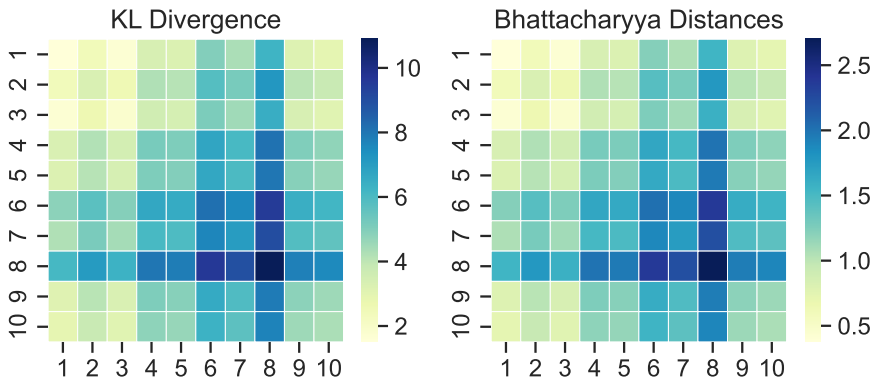


Figure 4.4 – Distance matrix of callers in WavLM's embedding space. The off-diagonal values represent the average inter-caller distances, while the diagonal entries the average intra-caller distances. Darker regions indicate higher dissimilarity.

4.4 Caller Detection Study

4.4.1 Classifiers

Table 4.2 – Search space to find optimal hyperparameters.

Classifier	Hyperparameters	Search space
RF	# Estimators	[50, 500, 1000, 2000]
	Max # Features	['auto', 'sqrt', 'log2']
	Criterion	['gini', 'entropy']
	Min samples leaf	[1, 2, 4]
AB	Learning rate	[0.1, 0.2, 0.5, 1]
	Algorithms	[SAMME, SAMME.R]
	Max # Estimators	[50, 500, 1000, 2000]
SVM	C	1e[-5, -4, -3, -2, -1, 0]
	Kernel	[RBF, Linear, Polynomial]
	Gamma	['scale', 'auto']
LSVM	C	1e[-5, -4, -3, -2, -1, 0]
	Max # Iterations	10000
	Class weights	['balanced', 'None']

Based on the insights of our caller discrimination analysis, we proceed to classify the statistics computed over the caller-groups for the task of caller detection in a 5 fold cross-validation (CV) framework. We concatenate the mean and variance of the Gaussians into a single functional vector, and use them as our fixed-length representations for classification.

We use Random Forest (RF), Ada Boost (AB), Support Vector Machines (SVM), and Linear SVM (LSVM) algorithms to classify the computed functional vectors. The difference between Linear

SVM and SVM with a linear kernel lies in the former's utilization of a squared hinge-loss, while the latter employs a regular hinge-loss.

To determine the most robust classification technique, we employ the grid search methodology with F1-Macro score as the optimization criterion, integrated into the Scikit-learn toolkit. We tune the hyperparameters for each fold, across the train and validation sets over the search space given in Table 4.2.

4.4.2 Evaluation Metrics

To evaluate the effectiveness of our proposed approach for the given task, we present the area under the curve (AUC) scores, which provide a evaluation of the performance of all the classifiers in correctly classifying the positive instances against negative. For SVM it is computed pairwise using a 'one-vs-one' methodology, while for the other classifiers it is calculated in a binary 'one-vs-rest' framework, by averaging the AUC scores for each class against all others.

4.4.3 Results and Discussion

Table 4.3 – Macro AUC scores [%] on Test with 5-fold CV for caller detection task using different classifiers.

Model	AB	LSVM	RF	SVM
APC	71.44	65.18	70.89	79.16
VQ-APC	71.60	65.58	70.04	78.45
NPC	72.61	66.27	71.50	77.32
Mockingjay	72.39	64.43	71.75	78.44
TERA	70.34	64.57	68.43	74.03
Mod-CPC	72.62	64.05	69.81	75.96
Wav2Vec2	74.41	63.94	70.18	75.85
Hubert	71.71	64.14	70.17	75.64
DistilHubert	70.77	65.11	70.34	76.26
WavLM	73.97	65.32	70.74	78.60
Data2Vec	69.81	62.58	68.23	73.04
Average	71.97	64.66	70.19	**76.61**

Table 4.3 summarizes the performance of the different classifiers on all the embedding spaces. The results show that SVM significantly outperforms the other classifiers across all embedding spaces. The decision tree-based ensemble methods, AdaBoost and Random Forest, exhibit comparable performance for most models, and consistently outperform Linear SVM. This suggests that the relationship between the features in the embedding space and their labels is likely to be complex and non-linear, which can be modelled by ensemble methods to some

degree, but not to the extent of non-linear SVMs.

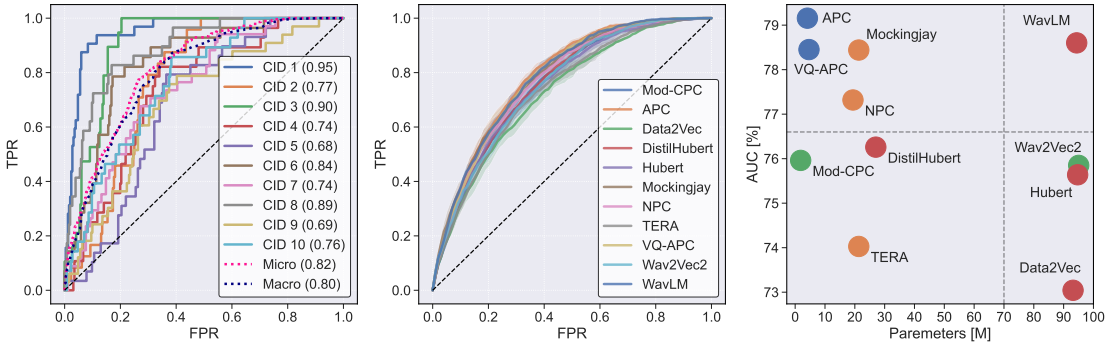


Figure 4.5 – a) ROC curves per caller class (CID) for WavLM embeddings using SVM on one fold of *Test*. b) Macro average ROC curves of all models on *Test* using SVM over all folds. Shaded areas represent ± 1 std over the k-folds. c) Model size against performance. Model pre-training objective denoted as: • Masked prediction. • Autoregressive reconstruction. • Contrastive • Masked reconstruction.

Figure 4.5a) shows the caller classification performance in distinguishing a positive class from the negative instances using SVM on a single *Test* fold. We can observe that all callers are systematically distinguished in this binary framework, including the classes with a low amount of data (CID 6–8).

Figure 4.5b) visualizes SVM's average performance for each embedding space across the 5 folds, with the shaded areas representing ± 1 std. The results clearly demonstrate that the embedding spaces of all models are capable of successfully differentiating Marmoset callers, indicating that SSL models pre-trained on human speech data can generate salient representations capable of distinguishing animal vocalizations regardless of the pre-training criterion.

Figure 4.5c) illustrates the relationship between the number of parameters and classification performance for each embedding space. The plot is divided into four quadrants to highlight differences in performance. Interestingly, WavLM's embedding space is found to be more separable than the other masked prediction models, indicating that its masked speech denoising task may be more effective in capturing animal caller identification information than Hubert's masked speech modeling. Surprisingly, both auto-regressive reconstruction based models perform exceptionally well with significantly fewer parameters. These findings suggest that while all pre-training criteria can yield competitive performance, some may be more efficient than others, allowing models with simpler architectures and fewer parameters, such as APC and AQ-APC, to perform comparably to larger models like WavLM. Finally, we observe that Data2Vec is not as successful as the other masked prediction based models, despite the same number of pre-training hours, corpus and comparable number of parameters. While it has shown to outperform the other masked prediction models in human speech, it seems to clearly learn weaker representations for the task of domain adaptation.

4.5 Conclusions

This section investigated the applicability of self-supervised representations, pre-trained on human speech through different approaches, to analyze vocalizations in the bioacoustics domain. To that end, we conducted and validated two lines of investigation on Marmoset calls in a caller detection framework.

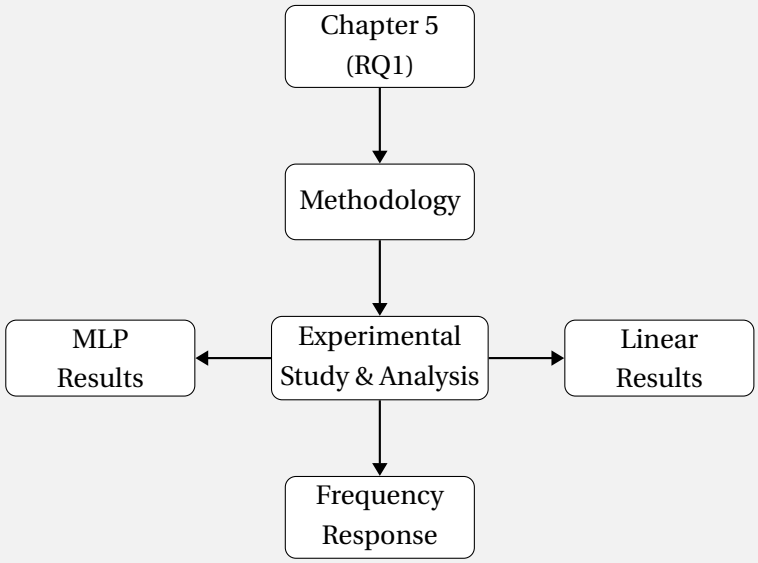
We first conducted a caller discrimination analysis study on the training data to examine the linear separability of eleven pre-trained embedding spaces by splitting the training data into caller-groups, and then calculating the intra-group and inter-group distances through a multivariate Gaussian distribution framework. The results showed that all spaces exhibited similar distance patterns, and that distinguishing marmoset callers is possible with a linear classifier but only to a certain extent.

For our second investigation, we conducted a caller detection study to analyze whether the embedding spaces of said caller-groups can be systematically distinguished by class. We trained four classifiers to predict the classes of the caller-groups in 5 fold cross-validation framework. The results show that we can effectively distinguish all Marmoset callers, including those with low data, in a binary classification framework. The results also show that non-linear SVMs are able to most accurately model the non-linear relationship between the features of the embedding space. Finally, we observe that although all embedding spaces seem effective at the caller detection task, some learning objectives may be more efficient than others.

In summary, our research demonstrates that self-supervised representations pre-trained on human speech can effectively classify vocalizations in the bioacoustics domain for tasks such as Marmoset caller detection, even without fine-tuning. These findings can greatly benefit bioacoustics researchers looking to distinguish individual identities within a specific species in their acoustic data. Additionally, we anticipate that further fine-tuning of these models on relevant bioacoustics downstream tasks can improve performance. Therefore, we plan to investigate the impact of model size on performance after fine-tuning, and also explore adapting the embedding spaces for other tasks like call-type classification in our future work.

5 Beyond Caller Identity: Decoding Marmoset Vocal Communication

Chapter Schematic Overview



Publication Note

The material presented in this section is adapted from the following publications:

- E. Sarkar et al. (2025). "On Feature Representations for Marmoset Voca Communication Analysis". In: *Bioacoustics*, pp. 1–15.

Collaboration Note

Part of this work was conducted in collaboration with other NCCR groups:

- WP *Accomodation*, project *cooperation*, group *social cognition of language*, led by Prof. Dr. J. Bukart (Department of Comparative Language Science, Institute of Evolutionary Anthropology, and Center for the Interdisciplinary Study of Language Evolution, UHZ).

5.1 Introduction

The advancements in human speech processing have accelerated and impacted research in non-human communication, such as bioacoustics, i.e. the study of animal sounds. However, in the existing works, there are three main limitations. First, most of the studies have been carried out on small datasets. Second, these studies have been conducted on datasets intended for specific scenarios. Due to a lack of validation, it is unclear whether the methods studied on one dataset would scale to another. Third, there is limited prior knowledge about what type of information is relevant for different call analysis tasks. There is a need to overcome these limitations to advance the development of automatic analyses of marmoset vocalizations. Chapter 4 addressed this gap through a proof-of-concept study on a single dataset and a binary caller detection task. The present chapter extends that investigation with a specific focus on feature representations for automatic marmoset call analyses, where we investigate three prominent feature representation methods, namely, (a) hand-crafted features, (b) self-supervised learning-based representations, and (c) end-to-end acoustic modeling, on three different marmoset call datasets and three different tasks (call type, caller identity, and caller sex classification).

This chapter is organized as follows. Section 5.2 presents the different datasets, tasks, and investigated feature representations. Section 5.3 and 5.4 present the studies and analysis of the results respectively. Finally Section 5.5 concludes the chapter.

5.2 Methodology

5.2.1 Datasets and Tasks

We conduct investigations on three different marmoset datasets, namely the InfantMarmosetsVox (IMV), Bosshard, and Wiercka datasets, denoted in this chapter as D_1, D_2, and D_3, respectively. D_2 and D_3 contain vocalizations produced by adult individuals, while D_1 originates from infant marmosets (Sarkar and Magimai.-Doss, 2023). Consequently, D_1 is expected to encompass different call types, likely characterized by higher frequencies compared to those in D_2 and D_3. Furthermore, D_2 and D_3 are gathered from the same colony, while D_1 was obtained from a different one. All the datasets consist of audio recordings of marmosets vocalizations segments, collected and hand-labeled with the start and end time by experienced researchers. In addition to call-type and caller identity annotations of each vocalization provided for all three datasets, D_1 and D_2 also include information about the sex of the vocalizing individual. For more details regarding the datasets, the reader is referred to 3.

We discard any segments labeled as 'silence' and 'noise', and only keep the vocalization segments. The log distribution of the vocalization lengths of the three datasets is presented in Figure 5.1. We can observe that D_1 has the shortest median vocalization length at 127 ms, with D_2 and D_3 at 175 and 1037 ms respectively. Based on the given annotations, we define multi-class tasks, specifically call-type, caller, and sex classification, henceforth referred to as

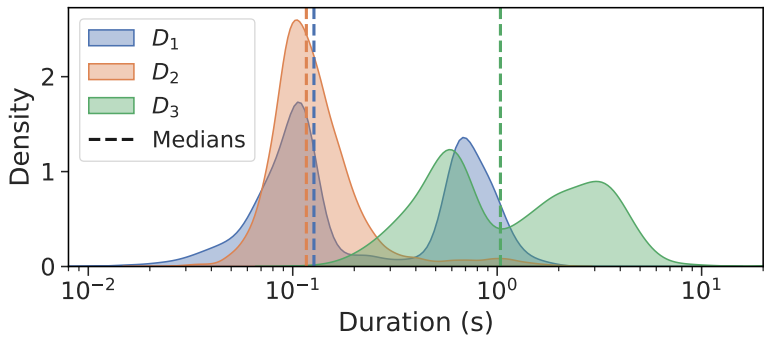


Figure 5.1 – Log distribution of vocalization lengths per dataset. The medians are calculated over the entirety of each dataset.

CTID, CLID, and SID respectively. Table 5.1 gives the number of vocalization segments S, their total duration length L, the native sampling rates, as well as the number of classes n_c for each task across datasets.

Table 5.1 – S indicates the number of data samples, L the sum of all vocalizations segment durations (in minutes), and SR the native sampling rate of the given data (kHz). n_{task} is the number of classes of each task-dataset permutation.

$\mathcal{D}$	S	L	SR	n_{CTID}	n_{CLID}	n_{SID}
D_1	73K	464	44.1	11	10	-
D_2	14K	37	300	7	8	2
D_3	5K	138	125	12	8	2

5.2.2 Feature Representations

We investigate the following feature representations:

1) *Hand-crafted features*: Highly Comparable Time-Series Analysis (HCTSA) is an interpretable signal processing-based framework that has been demonstrated to be useful for diverse time series application domains (Fulcher, Little, and Jones, 2013). In this framework, a set of 7700 features are extracted by characterizing the signal by different time series analysis methods, such as, linear correlation, modeling fitting (e.g., autoregressive moving average analysis, GARCH), wavelet analysis, extraction of information theoretic measures, which then is combined with feature selection to build statistical models for the end task. In the literature, these features have been investigated for behavioural birdsong discrimination (Paul et al., 2021), automated acoustic monitoring of ecosystems (Sethi, 2020), as well as marmoset caller identification (Phaniraj et al., 2023). One of the challenges of HCTSA approach is computational complexity and involves an evaluation of many similar features. In a recent work, CAnonical Time-series CHaracteristics (Catch22) features, a subset of the HCTSA feature set has been proposed which exhibit a strong performance across 93 real-world time-series

classification problems, but are also minimally redundant (Lubba et al., 2019). In this work, we investigate the Catch22 features, denoted as C22.

2) *Pre-trained self-supervised learning (SSL) based features*: Inspired from the recent study presented in (Sarkar and Magimai.-Doss, 2023), we investigate the use of feature representations extracted from pre-trained SSL neural networks trained on human speech for marmoset call analysis. We extend the investigations from caller detection to call type, caller ID and sex classification. Furthermore, contrary to the previous work (Sarkar and Magimai.-Doss, 2023), which focused only on the last transformer layer representation, in this work we investigate representations obtained from all the transformer layers to gain insight which level of layer representations are informative for marmoset call analysis.

3) *End-to-end acoustic modeling*: With advances in deep learning, acoustic modeling approaches have emerged in speech and audio processing where raw signal can be modeled to learn task-dependent information from the signal in an end-to-manner with minimum prior knowledge (Palaz, Collobert, and Magimai-Doss, 2013; Trigeorgis et al., 2016; Zazo et al., 2016; Muckenhirn, Magimai.-Doss, and Marcel, 2018). Such approaches hold potential for advancing marmoset call analysis, as they could help not only in addressing the lack of reliable task-dependent prior knowledge challenge, but also in gaining insight into the task relevant acoustic information learned by such trained networks through analysis (Muckenhirn, Magimai.-Doss, and Marcel, 2018; Muckenhirn et al., 2019; Palaz, Magimai.-Doss, and Collobert, 2019). The insight gained could then be further validated through linguistic studies. Motivated by these aspects, we investigate this approach.

A sub-challenge that arises when analyzing marmoset calls is the range of frequency information to be modeled. More precisely, the fundamental frequencies (typically corresponding to the peak frequency) of adult marmoset vocalisations span a range of 6-13 kHz, depending on the *call-type* (J. A. Agamaite et al., 2015). However, as can be seen in Table 5.1, datasets are collected at varying sampling frequencies. Furthermore, the SSL neural networks are typically pre-trained on speech signal of 8 kHz bandwidth (i.e., 16 kHz sampling frequency). As part of the investigation, we thus also study the impact of sampling rate (SR) on marmoset call analysis tasks.

5.3 Experimental Study

5.3.1 Systems

For each task, we divided all datasets into training, validation, and test sets, named *Train*, *Val*, and *Test* respectively, following a 70:20:10 split ratio, in order to train models on a sufficiently large number of samples, while ensuring sufficient data points for model evaluation and validation. *Train* is used to train the models, *Val* to tune any hyperparameters, and *Test* to evaluate the trained models on unseen data. We then developed the following systems for each task on each dataset to investigate the aforementioned feature representations:

1) We used *pycatch22* to extract a feature vector $\boldsymbol{x} \in \mathbb{R}^{1\times D}$ for each utterance, where $D = 24$, and feed it to a simple, non-linear multilayer perceptron (MLP). We implement three blocks of [Linear, LayerNorm, ReLU] layers, with 128, 64, and 32 number of hidden units respectively, followed by a final linear layer to obtain the posterior probabilities. The classifier is trained for 30 epochs, using a batch size 16 and learning rate $\eta = 1e-3$.

2) As it is challenging to investigate all the different types of pre-trained SSL feature representations across all tasks and datasets, we simply chose WavLM (S. Chen et al., 2022), as it was found to yield strong performance on the task of marmoset caller detection (Sarkar and Magimai.-Doss, 2023), been found to scale well to different human speech processing tasks in the SUPERB challenge (S.-w. Yang et al., 2021). For each layer, we extracted frame-by-frame variable-length feature representations $\boldsymbol{x} \in \mathbb{R}^{N\times D}$, where $D = 768$ and N the variable number of frames (contingent on the vocalization length). We then converted these embeddings into utterance-level fixed-length representations $\boldsymbol{f}_{\mu\sigma} \in \mathbb{R}^{1\times 2D}$ (denoted as WLM), by computing and concatenating the first and second order statistics across the frame axis on the extracted features. An MLP of same three layer architecture as *C22* is then trained with the fixed length feature as input.

3) We trained a convolutional neural network (CNN) based end-to-end acoustic modeling system (denoted as E2E) that takes a raw waveform as input and classifies to the output classes. Following the literature in speech processing (Dubagunta, Vlasenko, and Magimai.-Doss, 2019; Nallanthighal et al., 2021; Purohit et al., 2023), the E2E system consists of four convolution layers followed by an adaptive pooling layer and two hidden layers. The E2E system is optimized with a cross-entropy cost function with an early stopping criteria.

Table 5.2 – CNN model parameters. n_f denotes the number of filters, n_{hu} the the number of hidden units, and σ the activation function.

Layer	$\boldsymbol{kW}$	$\boldsymbol{dW}$	$\boldsymbol{n_f / n_{hu}}$	**Padding**	$\boldsymbol{\sigma}$
Conv 1	kW	dW	128	-	ReLU
Conv 2	10	5	256	-	ReLU
Conv 3	4	2	512	2	ReLU
Conv 4	3	1	512	1	ReLU
Adapt	-	-	-	-	-
FC 1	-	-	512	-	ReLU
FC 2	-	-	256	-	ReLU
FC 3	-	-	n_c	-	-

Table 5.2 presents the architecture of the E2E system. The first convolution layer kernel width kW and shift dW was chosen based on the sampling frequency. More precisely, based on the understanding gained from speech studies, we chose those hyper-parameters to strike a balance between the length of the convolution filter and enough pitch cycles being modeled (Muckenhirn, Magimai.-Doss, and Marcel, 2018). For 44.1 and 60 kHz sampling frequency, we chose $kW = 1$ ms and $dW = 0.05$ ms, respectively. As marmoset calls have fundamental

frequency around 5 kHz and above (J. A. Agamaite et al., 2015), 1 ms signal would be expected to contain around 10 pitch cycles or more. However, for 16 kHz sampling frequency, 1 ms would contain only 16 samples, i.e. at the most 1-2 sample(s) representing each pitch cycle. This may not hinder capturing the pitch frequency information in the marmoset call well. So, for 16 kHz we set $kW = 10$ ms and $dW = 0.5$ ms. The training batch size 16 and learning rate of 0.001, same as the MLP classifier for C22 and WLM. The optimization configuration simply consisted of Adam and a dynamic learning rate scheduler which reduces the learning rate η when the selected optimization criterion, in this case *Val* UAR, shows no improvement after 10 epochs.

In the case of *C22*, we developed systems at native sampling frequency and downsampled acoustic signals: 16 kHz for D_1, 60 and 16 kHz for D_2, and 60 and 16 kHz for D_3. In the case of WLM, we developed systems with signals downsampled to required pre-training sampling rate of 16 kHz. For E2E system, D_2 and D_3 signals were downsampled to 60 and 16 kHz. To evaluate the systems we used Unweighted Average Recall (UAR) as the metric to account for any class imbalance.

5.3.2 Results

Table 5.3 shows the performances of systems based on different feature representations. For the sake of clarity, only the best layer and worst layer performances are reported for WLM. Figure 5.2 presents the layer-wise performances for all tasks on all datasets for WLM. Note that layer 0 corresponds to the output embedding of the CNN encoder, where as the other 12 refer to the outputs of the transformer encoder layers. The performances are all above chance level, i.e. $100/n_c$, for all systems.

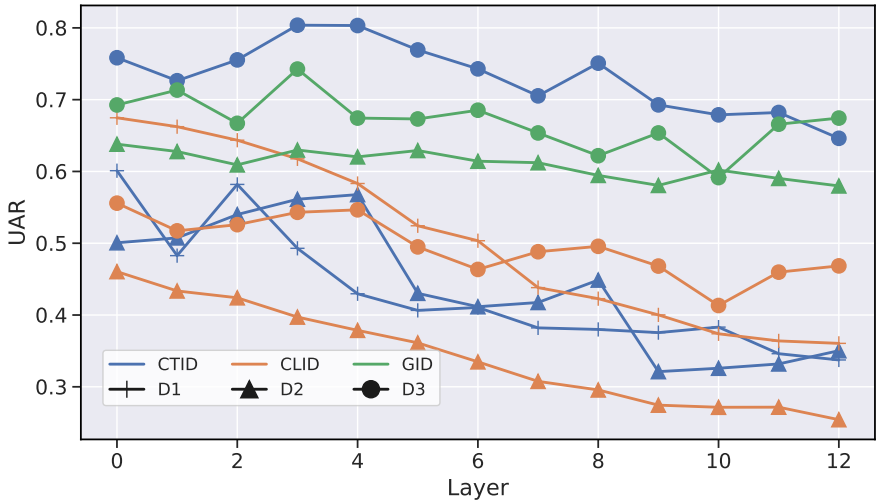


Figure 5.2 – Layer-wise UAR scores for WLM for all tasks and datasets. The layers follow the same indexing as (S. Chen et al., 2022).

Ignoring the sampling frequency aspect, it can be observed that E2E yields the best perfor-

Table 5.3 – UAR scores on *Test* on features $\mathcal{F}$. WavLM's best and worst layer's score is given. For each dataset, the best score across features is bolded per task.

$\mathcal{D}$	$\mathcal{F}$	SR	CTID	CLID	SID
D_1	C22	44.1	51.04	47.58	N/A
		16	37.72	34.54	N/A
D_1	WLM	16	60.10	67.47	N/A
			33.74	36.05	N/A
D_1	E2E	44	**68.32**	**74.12**	N/A
		16	53.03	59.94	N/A
D_2	C22	300	37.68	43.56	**66.24**
		60	32.50	35.52	63.38
		16	35.65	35.32	58.14
D_2	WLM	16	**56.77**	46.05	63.80
			32.11	25.42	57.98
D_2	E2E	60	42.03	**49.78**	62.36
		16	37.65	36.21	60.15
D_3	C22	125	64.32	43.19	62.80
		60	65.67	45.50	61.22
		16	52.59	39.43	57.32
D_3	WLM	16	**80.38**	**55.58**	**74.26**
			64.62	41.33	59.14
D_3	E2E	60	65.31	47.92	60.73
		16	66.24	31.31	56.59

mances for D_1's CTID and CLID tasks. For D_2, WLM yields best performance for CTID, E2E for CLID, and C22 for SID. On both D_1 and D_2, we can observe that WLM yields competitive systems, however in the case of D_3, WLM's third layer representations consistently yield the best performance across all the tasks (see Figure 5.2), and outperform C22 and E2E. Although WLM yields competitive performances on D_1 and D_2, it is difficult to systematically compare to C22 or E2E as different layers yield best performance for different tasks.

Furthermore, it can be observed that the 16 kHz SR performance is generally inferior across different datasets and tasks for C22 and E2E. This finding is in line with the understandings in the literature gained by analysis of different call types which showed that most marmoset call types extend into frequencies above 8 kHz (J. A. Agamaite et al., 2015). This implies that, with an 8 kHz bandwidth, certain vital information for specific call types might be lost, rendering it increasingly challenging, if not impossible, for the classifier to accurately categorize certain calls. Indeed, it can be observed that C22 systems yield superior performance with the native SR compared to 16 kHz for all datasets. This emphasizes that higher frequencies are likely to

contain valuable information. A comparison between C22, WLM and E2E at 16 kHz sampling frequency demonstrates the potential of SSL based feature representations learned on human speech.

It is worth noting that a recent, independent study explored representations learned from other acoustic domains such as general audio, which includes audio event classes such as environmental sounds, musical instruments, and human and animal vocalizations. They demonstrated on D_1 that increasing the pre-training bandwidth of a PANN model (Kong et al., 2020), pre-trained on the AudioSet dataset with log-mel spectrogram inputs, improved performance on both CTID and CLID tasks (Sarkar and Magimai.-Doss, 2024). However, the study didn't explicitly disentangle whether these improvements resulted from the increased bandwidth itself, the spectrogram-based inputs, or from the inclusion of some animal vocalizations in the pre-training dataset. This distinction still remains an important open question for future investigations.

5.4 Analysis

5.4.1 Layer-wise Linear Performance Analysis

In Figure 5.2, it can be observed that lower layer representations tend to yield better systems. To further ascertain that, we carried out layer-wise classification performance of the same tasks using a simple linear classifier (single layer perceptron). Figure 5.3 shows the results independently normalized per-task to a [0, 1] range. It can be observed that the lower layers are much more salient representations for all three tasks across all datasets when compared to higher layers. A possible explanation is that, because WavLM's CNN encoder operates directly on the raw waveform, the early layers capture fundamental *acoustic* features and can leverage spectro-temporal variations relevant to tasks such as speaker identification and verification (S. Chen et al., 2022). Thus, these lower layers inherently generalize better to other acoustic domains, such as marmoset vocalizations. In contrast, the later layers – shown to perform well on *linguistic* tasks, such as speech or phoneme recognition – appear more specialized for human speech and consequently much less transferable to bioacoustics, resulting in lower performance. We can also observe that there is no consistent optimal layer for each task type across the datasets.

5.4.2 Frequency Response of Learnt Convolution Filters

We analyzed the frequency response of the first learned convolution layer filters of E2E systems by estimating the cumulative frequency response F_{cum} as (Palaz, Magimai.-Doss, and Collobert, 2019):

$$F_{cum} = \sum_{k=1}^{n_f} \frac{F_k}{\|F_k\|_2}, \tag{5.1}$$

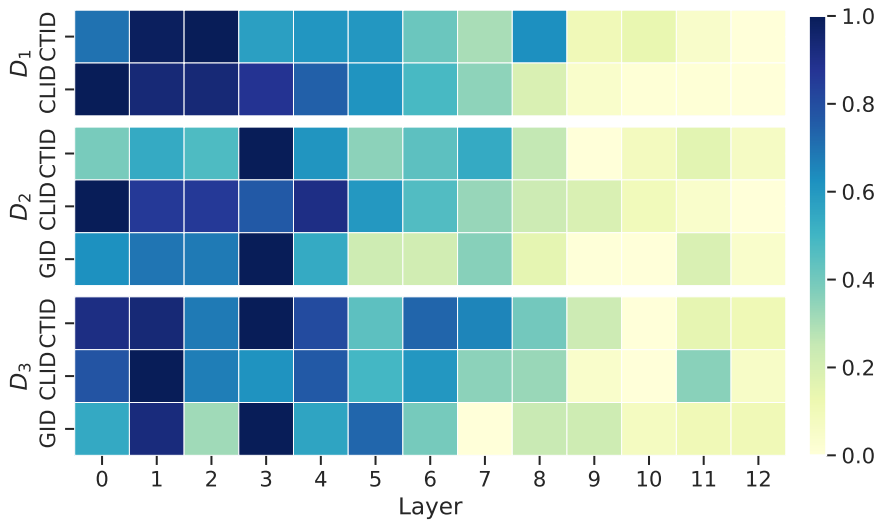


Figure 5.3 – Layer-wise UAR scores of WLM features modeled by single layer perceptron. The scores are normalized independently per task. Darker regions indicate higher performance.

where n_f denotes the 128 filters in the first convolution layer and F_k denotes discrete Fourier transform of filter k over 2048 DFT points.

Figure 5.4 shows the cumulative frequency response for each task per dataset at an SR of 16 kHz, and 44.1 or 60 kHz. With a 8 kHz bandwidth (left half), it can be observed that the emphasis is on frequencies 4-5 kHz and above irrespective of the task. As the bandwidth of the signal is increased (right half), it can be observed that emphasis is also given to higher frequency regions such as around 10 kHz or above. These observations further corroborate previous findings that most marmoset calls occupy frequency ranges beyond 8 kHz (J. A. Agamaite et al., 2015), and also explain the improved performance obtained with higher bandwidth signals. In addition, we observe that for different tasks the learned filters give emphasis to different frequency regions. A detailed analysis of the spectral information learned is part of our future work. Taken together, the analysis indicates that the E2E framework inspired from speech processing can be scaled to marmoset call analysis.

5.5 Conclusions

This chapter explored different feature representations or learning methods, namely hand-crafted feature Catch22, SSL feature representation WLM, and end-to-end acoustic modeling (E2E) for analyzing marmoset calls. Our investigations on three different datasets demonstrate that end-to-end acoustic modeling and SSL feature representations yield better systems than handcrafted Catch-22 features for call-type classification and caller identification, while also achieving comparable performances for sex identification at a common sampling rate. As a by-product, our studies demonstrated that (a) the utility of pre-trained SSL models on human speech can be extended to call-type and sex, besides caller discrimination and (b) end-to-end

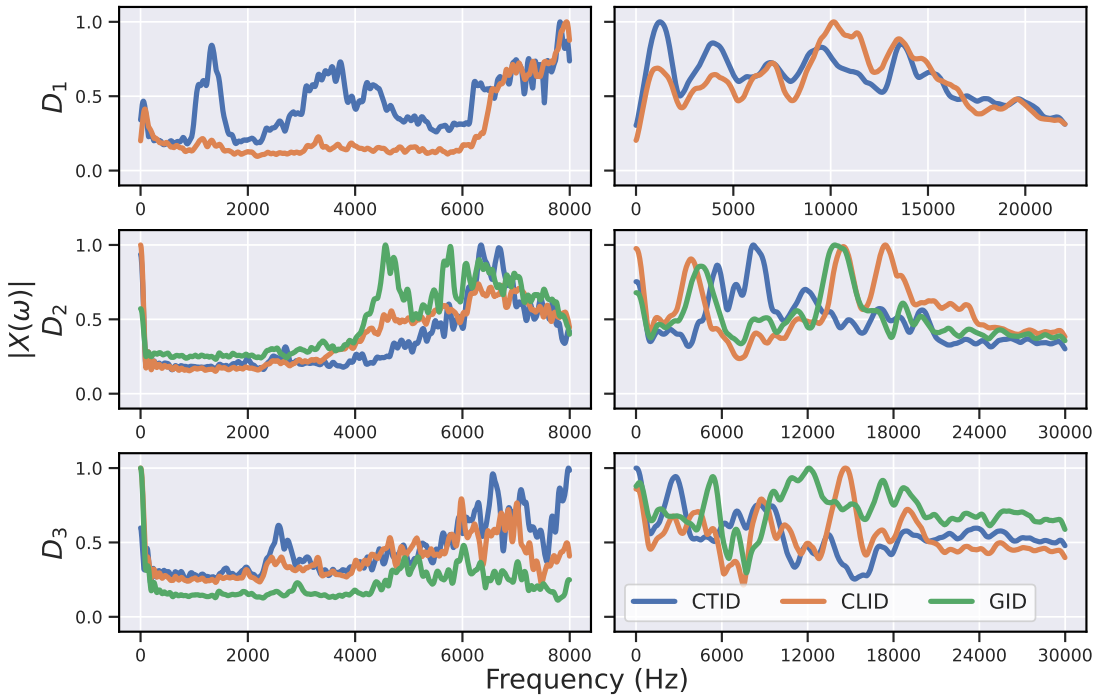
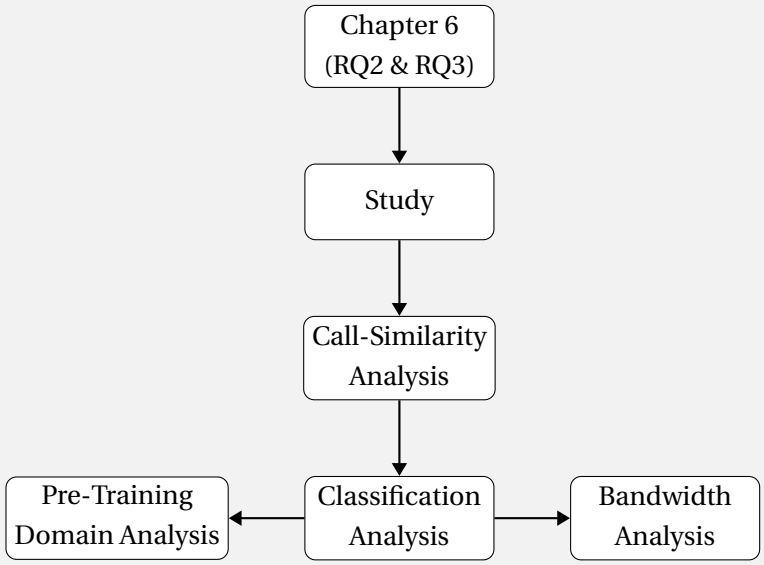


Figure 5.4 – Cumulative frequency response per task on all datasets. Sampling rate: 16 kHz (left), and 44.1 or 60 kHz (right).

acoustic modeling methods developed for speech processing can be scaled for marmoset call analysis. Our study raises a few pertinent questions such as: (a) with limited signal bandwidth how are SSL features informative about marmoset calls? (b) what kind of task specific spectral information is learned by the E2E systems?, and (c) how to combine the different approaches for improving marmoset call analysis? Furthermore, in this work we only investigated feature representations that directly modeled the raw input waveform. However, recent bioacoustic studies on bats, birds, and rodents have leveraged spectrogram-based methods (Goffinet et al., 2021; Ruff et al., 2020; K. R. Coffey, Marx, and Neumaier, 2019; N. Gu et al., 2024). Whether such approaches can offer distinct advantages over the waveform-based methods for marmoset vocal communication analysis remains to be determined. Our future work will investigate these questions.

6 Bandwidth Limitation in Speech and Audio SSL Models

Chapter Schematic Overview



Publication Note

The material presented in this section is adapted from the following publications:

- E. Sarkar and M. Magimai.-Doss (2024). "On the Utility of Speech and Audio Foundation Models for Marmoset Call Analysis". In: *4th International Workshop on Vocal Interactivity In-and-between Humans, Animals and Robots (VIHAR2024)*.

Supplementary Material

- **Source Code**: https://github.com/idiap/speech-utility-bioacoustics.

6.1 Introduction

Chapter 4 and 5 demonstrated that neural representations derived from models pre-trained on human speech through self-supervised learning (SSL) could distinguish individual marmoset call-types and caller identities (Sarkar and Magimai.-Doss, 2023; Sarkar et al., 2025). We argued that SSLs only learn the intrinsic structure of the unlabeled input signal, typically through a masking-based pre-text training task, to capture essential information independently of any domain-specific knowledge, such as human speech production, and thus can be cross-transferred across different acoustic domains, such as bioacoustics. Building on these findings, this chapter investigates the utility and limitations of such pre-trained SSL models for the purpose of marmoset call analysis, with a focus on the following key points:

1. **Bandwidth**: Given that these models are typically pre-trained on human speech with a bandwidth of 8 kHz, we address their mismatch with the biological vocalization and auditory range of marmosets, predominantly concentrated in the 5–10 kHz spectral region (Osmanski et al., 2016), and thus evaluate their capability to accurately represent marmoset calls. By examining models pre-trained across varying bandwidths, we aim to evaluate their effectiveness in adequately representing marmoset calls, and seek to clarify how model bandwidth influences their classification.
2. **Pre-training domain**: It remains unclear how models pre-trained on human speech compare to trained on other acoustic domains for accurately capturing marmoset call characteristics. We examine representations produced by different pre-training sources, such as human speech and general audio, across supervised and self-supervised learning frameworks, against a spectral baseline to identify the most suitable pre-training source for cross-domain bioacoustic signal analysis.

6.2 Methodology

6.2.1 Dataset and Tasks

For our study, we used the InfantMarmosetsVox (IMV) dataset (Sarkar and Magimai.-Doss, 2023), which contains 72,921 labeled marmoset vocalization segments (totaling to 464 minutes), sampled at 44.1 kHz, across ten marmoset individuals and contains eleven marmoset call-types. Table 6.1 presents the data distribution in function of the call-types and callers. For our experiments, we divide the dataset into a *Train*, *Val*, and *Test* sets, following a random 70:20:10 split. We denote call-type and caller identity multi-class classification as CTID and CLID respectively.

Figure 6.1 gives the visualizations of all call-types as well the density distribution of the spectrums across the entire dataset. Frequencies below 500 Hz are nullified purely for visualization to eliminate any low-frequency noise. We can observe that information starts at around 7-8 kHz for most calls in this dataset.

Table 6.1 – InfantMarmosetsVox dataset statistics.

ID	Call-type	Count	Caller ID	Count
0	Peep (pre-phee)	1283	0	15521
1	Phee	27976	1	8648
2	Twitter	36582	2	13827
3	Trill	1408	3	5838
4	Trillphee	728	4	5654
5	Tsik Tse	686	5	3522
6	Egg	1676	6	4389
7	Pheecry (cry)	23	7	2681
8	TrllTwitter	293	8	6387
9	Pheetwitter	2064	9	6454
10	Peep	202	-	-
	Total	**72921**	**Total**	**72921**

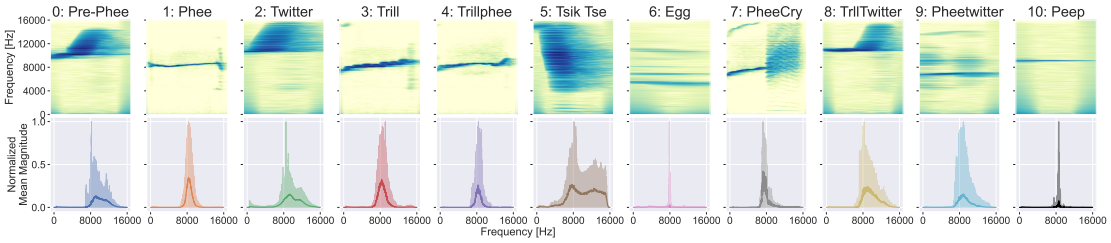


Figure 6.1 – Marmoset vocalizations with a 16 kHz bandwidth. Top: Spectrograms of a single call-type vocalization. Bottom: The mean spectrum for all vocalizations per call-type across the dataset, normalized. Shaded areas indicate ± 1 std from the mean spectrum.

6.2.2 Models and Feature Representations

For our study, we select four distinct frameworks for feature representations $\mathcal{F}$: hand-crafted (HC) features derived through signal processing techniques, neural representations obtained via self-supervised learning (SSL), pre-trained on either human speech or general audio, and features generated through supervised learning (SL) models pre-trained on general audio. These frameworks are summarized in Table 6.2. We extract the features from these frameworks by giving the marmoset calls as input.

Hand-crafted: The Highly Comparable Time-Series Analysis (HCTSA) framework, used for interpreting diverse time series data, extracts 7700 features through signal processing methods, such as LPC (Fulcher, Little, and Jones, 2013). It has been applied to diverse tasks such as birdsong discrimination (Paul et al., 2021), ecosystem monitoring (Sethi, 2020), and marmoset caller identification (Phaniraj et al., 2023). Despite its broad applicability, HCTSA's computational demands and feature redundancy are significant limitations. The CAnonical Time-series CHaracteristics (Catch22/C22), a streamlined subset of HCTSA, provides high performance with minimal redundancy across numerous classification problems (Lubba et al., 2019). We

Table 6.2 – # Parameters P and feature dimension D of selected models, pre-trained on AudioSet (AS) or LibriSpeech (LS).

$\mathcal{F}$	Corpus	P	D	Type
C22 (Lubba et al., 2019)	-	-	24	HC
WavLM (S. Chen et al., 2022)	LS	94.38M	1536	SSL
BYOL (Niizumi et al., 2021)	AS	5.32M	2048	SSL
PANN (Kong et al., 2020)	AS	8.08M	2048	SL

extend this feature set to a final dimension of $D = 24$ by appending the first and second order statistics, and use it as our spectral baseline.

SSL pre-trained on human speech: Following the approach in (Sarkar and Magimai.-Doss, 2023), we use feature representations from SSL models trained on human speech, extending it to both call-type and caller identity classification. We select the WavLM base model, pre-trained on the 960-hour LibriSpeech dataset, based on its effectiveness in marmoset call detection as well as its versatility in speech processing tasks as demonstrated in the SUPERB challenge (S.-w. Yang et al., 2021). For each layer, feature representations of length 768 are extracted for each frame. Then, they are transformed into fixed-length utterance-level representations by computing and aggregating first and second order statistics across the frame-axis, resulting in a final representation of length $D = 1536$.

SSL pre-trained on general audio: Expanding marmoset call analysis literature, we utilize embeddings from models pre-trained on the AudioSet (AS) dataset, which includes audio event classes such as environmental sounds, musical instruments, and human and animal vocalizations. Specifically, we choose the *AudioNTT2020* model from the BYOL-A architecture (Niizumi et al., 2021), extracting embeddings from its final fully connected layer of length $D = 2048$. Inputs are processed into log-mel spectrograms, adhering to the spectral parameters detailed in the original study, i.e. a 8 kHz bandwidth, 64 ms window size, 10 ms hop size, and 64 mel bins spanning from 60 to 7800 Hz.

SL pre-trained on general audio: We further investigate feature extraction from large-scale networks pre-trained for general audio pattern recognition. The *CNN14* model from the *PANN* network (Kong et al., 2020) is chosen, with pre-trained weights applied at three different bandwidths: 4, 8, and 16 kHz. This model employs a balanced sampling strategy across AudioSet's sound classes and also processes input vocalizations into spectrograms to extract log-mel filterbanks. For a bandwidth of 16 kHz, window and hop sizes are set to 1024 and 320 samples, respectively, and proportionally halved for 8 and 4 kHz. The model utilizes 64 mel bands, spanning from 50 Hz and to the Nyquist frequency. Embeddings of length $D = 2048$ are extracted from the linear layer preceding the final classification layer.

6.3 Call Similarity Analysis

This section presents a pairwise similarity analysis of the selected features on the *Train* set to identify any discernible patterns or correlations for given the vocalizations. Specifically, we investigate how variations in the bandwidth of the pre-trained models affect the similarity distribution of intra-class embeddings, and examine any distinctions between models pre-trained on speech against general audio. To compare the features, which are high-dimensional vectors, we use the cosine distance defined as $\text{sim}(\boldsymbol{x_1}, \boldsymbol{x_2}) = 1 - (\boldsymbol{x_1} \cdot \boldsymbol{x_2} / \|\boldsymbol{x_1}\| \cdot \|\boldsymbol{x_2}\|)$, bounded in $[0,2]$. Two features are identical when their cosine distance is 0, orthogonal at 1, and opposite at 2. For WavLM, we select the first layer, and only use the first half of the extracted features, corresponding to the mean values averaged frame-wise.

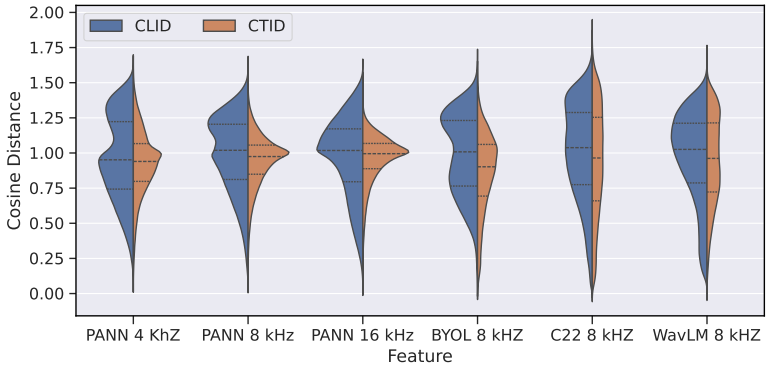


Figure 6.2 – Distribution of pairwise cosine distances.

Figure 6.2 presents the overall distribution of pairwise distances. The distributions are overlapping, centering around a median distance of 1 for all representations, suggesting a lack of clear correlation or similarity within the embeddings generated. Figure 6.3 further delineates the distributions into distance matrices for each feature set, where diagonal and off-diagonal entries correspond to intra-class and inter-class distances respectively. In an ideal scenario, embeddings from the same call-type or caller would exhibit closer distances, where as embeddings from different classes would have a higher dissimilarity.

We can observe that the models pre-trained on general audio datasets (BYOL and PANN) yield more distinct peaks and diagonals, on figures 6.2 and 6.3 respectively, compared to those pre-trained on human speech (WavLM) or the handcrafted baseline (Catch22). This distinction is more pronounced for call-types than for caller identification. This is expected, given that the call-types are spread across caller classes (a caller produces different calls, while a call can come from any caller). Although these patterns indicate some level of class-specific clustering, the distribution of distances largely show that the features are highly orthogonal. The similarity analysis thus indicates minimal feature correlation, and suggests that classifying these vocalizations with a simple linear classifier would be challenging, as there is no clear linear separability between the classes.

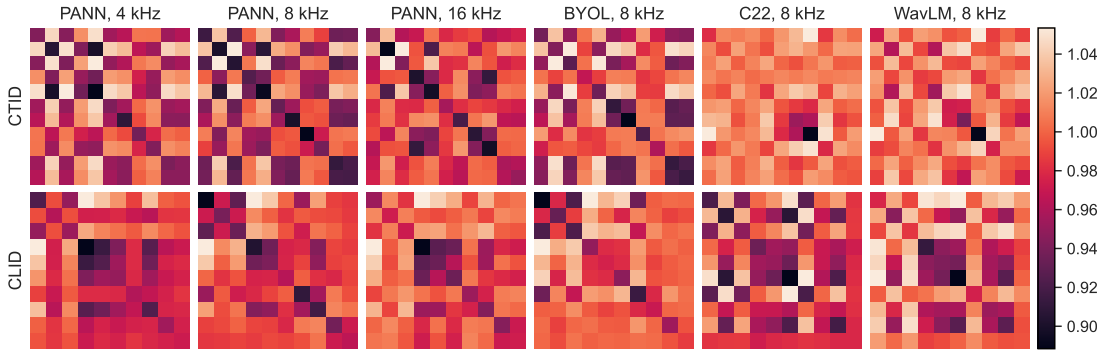


Figure 6.3 – Pairwise mean cosine distances matrices for features $\mathcal{F}$ at different bandwidths for call-types (CTID) and callers (CLID). Diagonal entries represent intra-class distances, and off-diagonal the inter-class. Darker regions indicate higher similarity.

6.4 Classification Analysis

Based on the insights of our similarity analysis, we aim to evaluate the saliency of the extracted representations, and proceed to classify them using a same non-linear MLP as in Chapter 4, for the multi-class classification tasks. We implement three blocks of [Linear, LayerNorm, ReLU] layers, with 128, 64, and 32 number of hidden units respectively, followed by a final linear layer to obtain the posterior probabilities. To evaluate the performance we used Unweighted Average Recall (UAR) as the metric to account for any class imbalance. To obtain robust results, we employ the grid search methodology with *Val* UAR score as the optimization criterion. We train the classifier for 30 epochs with cross-entropy loss, and search for the optimal hyperparameters values of η and batch-size across $2^{[5–9]}$ and [1e-3, 1e-4] respectively for each feature–task permutation on *Train* and *Val*. The optimization consists of Adam and a η-scheduler of factor 0.1 and patience of 10 epochs. Lastly, for WavLM, we classify each of the encoder layers [0–13] to identify the optimal layer.

Figure 6.4 presents the layer-wise scores for WavLM, normalized per task to a [0, 1] range. We can observe that the lower layers are clearly much more salient representations for both tasks compared to higher layers. Based on these results, we use the best individual WavLM layers for our two tasks.

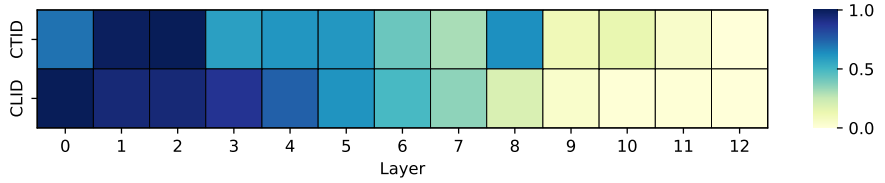


Figure 6.4 – Layer-wise UAR scores of WavLM features, normalized per task. Darker regions indicate a higher performance. Layer 0 corresponds to the output of the CNN encoder.

Table 6.3a) summarizes the classification results of the different feature sets at an 8 kHz

bandwidth (BW). Random performance is given as 100 over the number of classes. Notably, BYOL features outperform the other features, for both CTID and CLID, despite having fewer parameters than WavLM and PANN, while C22 proves to be the overall weakest representation. WavLM shows the highest difference in performance across tasks. Meanwhile, Table 6.3b) highlights the impact of pre-training bandwidth for salient representations on PANN features. The results clearly show that the bandwidth size correlates directly with the performance, increasing monotonically. Particularly, PANN features at 16 kHz achieve the highest performance across all features and BWs for CTID. BYOL embeddings at 8 kHz notably outperform PANN at 16 kHz for CLID. The best scores for both tasks are also closely matched in value.

Table 6.3 – UAR scores [%] on *Test* for pre-trained features $\mathcal{F}$. WavLM's best layer's score is given.

Section	$\mathcal{F}$	**BW**	**CTID**	**CLID**
(a)	Random	-	9.09	10
	C22	8	41.96	35.62
	WavLM	8	59.99	67.47
	BYOL	8	**63.64**	**68.30**
	PANN	8	58.54	56.02
(b)	PANN	4	46.27	41.10
	PANN	8	58.54	56.02
	PANN	16	**69.09**	**65.39**

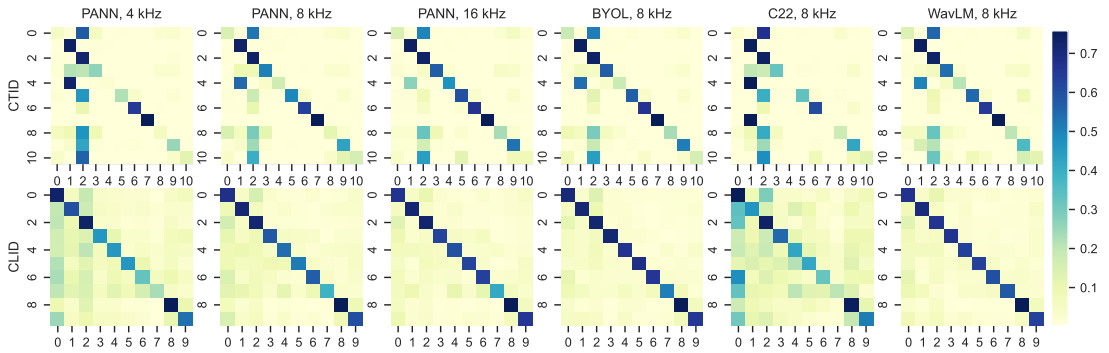


Figure 6.5 – Normalized confusion matrices with row indices representing true class labels. Darker diagonals signify higher performance.

Figure 6.5 shows the classifier's performance through confusion matrices. We can again clearly observe the monotonic improvement in CTID classification performance for PANN features as the bandwidth increases. We also notice a prevalent trend of false positives for call-type ID 2 (Twitter) across all feature sets, especially against IDs 0, 8, and 10, attributable to its high occurrence in the dataset and broad spectral range (Pistorio, Vintch, and X. Wang, 2006; J. Agamaite et al., 2015). The CLID results contain distinctly fewer misclassifications, which aligns with expectations since the call-types are spread among the different callers classes. The exception is C22, which yields the weakest performance. Caller classes with higher data

volumes (IDs 0 and 2) perform better compared to the others. Finally, a clear improvement in performance correlated with bandwidth is seen for PANN features, as with CTID.

6.5 Conclusions

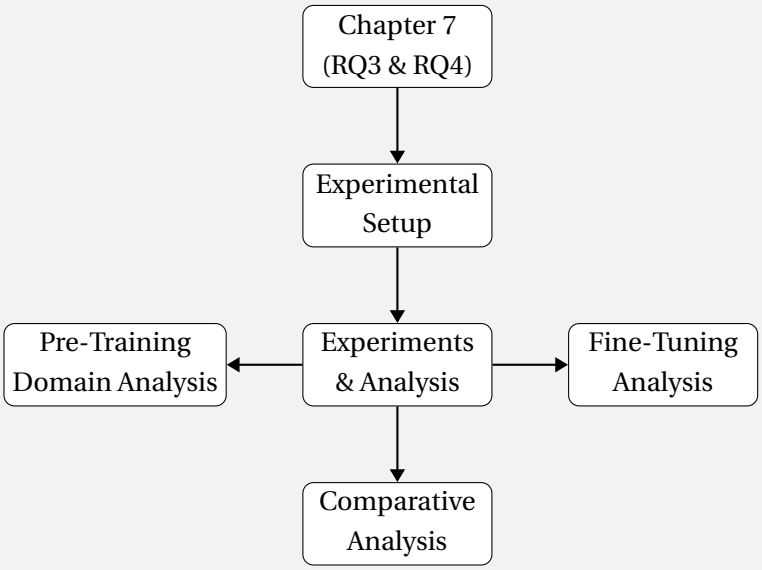
This chapter investigated the utility and limitations of foundations models, pre-trained on human speech or general audio. To that end, we conducted and validated two studies across two lines of investigation.

First we conducted a call similarity analysis, which revealed that the features extracted from these models lacked linear separability within or across classes. Then, we conducted a classification study which demonstrated that a non-linear classifier can still achieve substantial performance, and highlighted that a larger bandwidth directly correlates with improved performance. Classification of call-types also appeared to be more sensitive to bandwidth changes than caller identities. Additionally, the pre-training domain of speech and general audio showed comparable performances, with a distinct improvement over handcrafted features. Finally, we obtained close best performance for both call-type and caller classification tasks.

In conclusion, our findings underscore the potential of leveraging pre-trained SSL models for bioacoustic signals, particularly when the model's bandwidth aligns with the biological auditory and vocal range of the studied species. Future collaborative work with biologists and linguistics researchers could explore the biological implications of these results, especially in understanding the evolutionary aspects of marmoset vocal behaviour and their perceptual processing, to bridge the gap between computational models and biological insights in non-human vocal communication research.

7 Comparing Human and Non-Human Transference in SSL Models

Chapter Schematic Overview



Publication Note

The material presented in this section is adapted from the following publications:

- E. Sarkar and M. Magimai.-Doss (2025b). "Comparing Self-Supervised Learning Models Pre-Trained on Human Speech and Animal Vocalizations for Bioacoustics Processing". In: *Proc. of ICASSP*, pp. 1–5.

Supplementary Material

- **Source Code**: https://github.com/idiap/ssl-human-animal.

7.1 Introduction

Bioacoustics plays a crucial role in ecological and evolutionary research, providing insights into animal communication, biodiversity, and the origins of language. However, despite its significance, working with bioacoustic data presents several challenges: the data is often scarce, difficult to collect, noisy, and expensive to annotate. In recent years, advances in machine learning have made substantial progress in addressing these challenges (Stowell, 2022b). Notably, modern pre-trained deep learning foundation models have demonstrated impressive transferability to bioacoustic tasks, significantly advancing the field (Hagiwara et al., 2023b; Ghani et al., 2023; Dufourq et al., 2022; Heggan et al., 2024; Moummad, Farrugia, and Serizel, 2024). As demonstrated in Chapter 4, 5, 6, self-supervised learning (SSL) models pre-trained on *human speech*, in particular, have shown remarkable success in tackling various bioacoustic tasks, such as animal call-type classification (Sarkar and Magimai.-Doss, 2024; Mahmoud et al., 2024; Abzaliev, Perez-Espinosa, and Mihalcea, 2024; Heer Kloots and Knornschild, 2024; Shi, Itoyama, and Nakadai1, 2024), caller identification (Sarkar and Magimai.-Doss, 2023; Cauzinille et al., 2024; Knight et al., 2024), and species recognition (Hagiwara, 2023a). These models leverage large volumes of unlabeled data, prevalent in bioacoustics, by creating surrogate labels based on the intrinsic structure of the audio data, and then solving pre-text tasks designed to learn salient representations (A. Mohamed et al., 2022). Given the domain-agnostic nature of these pre-training tasks, SSL models have been effective in transferring from speech to bioacoustics without the need for domain-specific fine-tuning. Essentially, SSLs serve as powerful, general-purpose feature extractors for a wide range of downstream tasks.

Building on these developments, this chapter explores the following two points, aimed at analyzing SSLs for bioacoustics:

1. **SSL Pre-training Domain**: While SSL models pre-trained on human speech have shown strong transferability to bioacoustic tasks, recent research has explored pre-training on bioacoustic data itself, both in supervised and self-supervised frameworks (Kahl et al., 2021; Denton, 2023; Hagiwara, 2023a). The motivation behind pre-training on animal data is that these models may better capture species-specific vocal patterns and other properties unique to animal sounds. However, given that SSL pre-training is designed to learn general, domain-agnostic features, it is not yet clear whether pre-training directly on bioacoustics actually provides any significant advantage over SSLs pre-trained on human speech. Therefore, in this study, we systematically compare SSL models pre-trained on human speech against those on animal vocalizations, and evaluate their performance for bioacoustics processing across various datasets and tasks.

2. **Fine-tuning on Human Speech**: SSL representations have demonstrated strong performance on bioacoustic tasks without requiring fine-tuning, indicating their extracted latent representations can capture acoustically rich information capable of distinguishing animal call-types and caller identities. However, fine-tuning in a supervised framework

often forces the model to learn novel and more specialized patterns, such as phonetic distinctions and temporal structures, typically leading to further performance gains. As both human speech and animal calls encode structured vocal and linguistic information for communication, SSL models fine-tuned on *speech recognition* (ASR) may provide an additional inductive bias, enhancing the model's ability to recognize complex features in bioacoustic data. Therefore, we seek to explore whether fine-tuning pre-trained SSLs on human speech tasks, such as ASR, can further improve these models' capability to process animal vocalizations by capturing the subtle spectro-temporal characteristics present in animal calls, which may otherwise remain underrepresented in general SSL pre-training.

The rest of the chapter is organized as follows: Section 7.2 provides the experimental setup for the studies in this chapter, Section 7.3 presents and thoroughly analyzes the experiments' comparative results. Finally, Section 5.5 concludes the chapter.

7.2 Experimental Setup

7.2.1 Datasets, Tasks, and Protocols

We conducted the experiments for our studies on the three distinct bioacoustic datasets, summarized in Table 7.1. Figure 7.1 also presents a log distribution of their vocalization lengths.

Table 7.1 – L denotes the length [minutes], n_c the number of classes, SR the sampling rate [kHz], μ the median length [ms], σ the std.

Dataset	# Samples	L	SR	n_c	μ	σ
Watkins	1,697	295	–	32	1701	71245
IMV	72,920	464	44.1	11	127	375
Abzaliev	8,034	137	48	14	655	1313

Watkins (Sayigh et al., 2017): contains the recordings of different marine mammals, such as specific dolphins, whales, and seals. We chose Watkins for its multi-species vocalizations, rich acoustic variety, and high variance in segment lengths (Figure 7.1). It has been commonly used for bioacoustic benchmarking, particularly for evaluating modern deep learning models (Hagiwara, 2023a; Hagiwara et al., 2023b). We chose the 'best of' cut of the original dataset, a selected subset from the original 15,000 samples in total, deemed to be of higher sound quality and to contain less noise. The final dataset contains 1697 vocalization segments from 32 different species, totaling to 295 minutes, with a median length of 1701s. The sampling rate (SR) varies according to the recorded species.

InfantMarmosetsVox (IMV) (Sarkar and Magimai.-Doss, 2023): is an audio dataset of *Callithrix*

jacchus, a highly vocal new world primate. Marmosets were chosen for their complex social system, which allows them to encode vital information in their calls, such as identity, group affiliation, and dialect. They serve as surrogate models to understand the evolutionary origins of human vocal communication for neuro-biologists. The dataset consists of 72,920 segments representing 11 different call-types over 464 minutes. It was recorded from five pairs of infant marmoset twins, each recorded individually in sound-proofed rooms at 44.1 kHz SR, without communication with other marmoset pairs or the experimenters. The audio recordings were manually labeled by an experienced researcher. Although a large dataset by bioacoustics standards, each segment is predominantly short, with a median length of 127 ms. The spectral range of the calls is mostly centered around 7-8 kHz, although there is some information present above 16 kHz (Sarkar and Magimai.-Doss, 2024).

Abzaliev (Abzaliev, Perez-Espinosa, and Mihalcea, 2024): is a novel dog dataset (here referred to by the first author's name) consisting of 8,034 vocalizations from the v2017 Mescalina Bark ID dataset (Pérez-Espinosa et al., 2018). It contains 14 different call-types, ranging from normal, aggressive, fearful, and playful barks at strangers (IDs 0–3), to vocalizations related to owner interaction (4–5) and non-stranger/non-play sounds (6). It also contains positive or negative whines (7–8) and growls (9–10), barks associated with sadness or anxiety (11), and excitement upon the owner's arrival home (12). The recordings originate from various dog breeds, including Chihuahuas, French Poodles, and Schnauzers. The data was recorded at 48 kHz SR from a microphone, and followed a protocol designed and validated by experts in animal behavior. The dog vocalizations were induced by exposing the dogs to different types of external stimuli, with the participation of the owner and/or experimenter. We discard all the segments labeled as non-dog sounds, such as TV, cars, and appliances.

For our experiments, we divide the datasets into a *Train*, *Val*, and *Test* sets, following a random 70:20:10 split protocol.

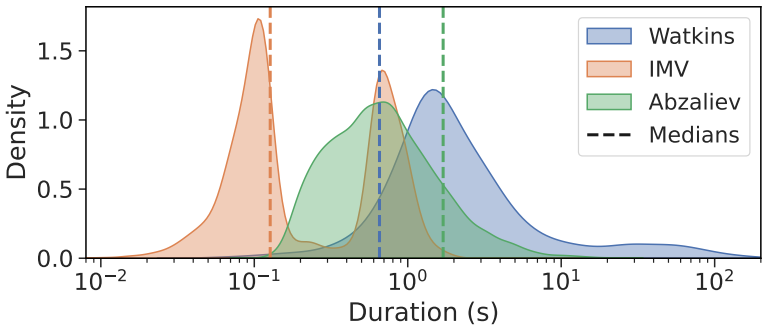


Figure 7.1 – Log distribution of vocalization lengths per dataset. The medians are calculated over the entirety of each dataset.

7.2.2 Models and Feature Representations

For our experiments, we select four different models to obtain our various feature representations $\mathcal{F}$. These consist of neural representations extracted through pre-trained (PT) models on animal vocalizations or human speech in a self-supervised learning framework, as well as their counterparts fine-tuned (PT+FT) in a supervised approach. The different features and their various properties are tabulated in Table 7.2.

Table 7.2 – # Parameters P [M] and feature dimension D of selected models. LS denotes LibriSpeech, AS represents AudioSet, and VVGS is VGGSound.

$\mathcal{F}$	**Corpus**	***P***	***D***	**TL**	**Type**
(Hagiwara, 2023a) AVES-Bio	FSD, AS, VVGS	94.68	768	12	PT
(W.-N. Hsu et al., 2021) HuBERT	LS 960	94.68	768	12	PT
(Baevski et al., 2020) W2V2	LS 960	95.04	768	12	PT
(Baevski et al., 2020) W2V2-100h	LS 960	95.04	768	12	PT+FT
(Baevski et al., 2020) W2V2-960h	LS 960	95.04	768	12	PT+FT
(S. Chen et al., 2022) WLM	LS 960	94.38	768	12	PT
(S. Chen et al., 2022) WLM-100h	LS 960	94.38	768	12	PT+FT

SSL pre-trained on animal vocalizations: We look at the AVES models family (Hagiwara, 2023a), which are essentially the same as HuBERT models, but pre-trained on bioacoustics data instead of human speech. We select them based on their effectiveness on numerous bioacoustic classification and detection tasks, as well as the extensive benchmarking. Although this model performs well compared to traditional classifiers (Hagiwara, 2023a), its performance has not been directly compared to a regular HuBERT model pre-trained on speech. The AVES set are pre-trained on combinations of publicly available audio datasets, namely FSD50K (Fonseca et al., 2021), AudioSet (Gemmeke et al., 2017), and VGGSound (H. Chen et al., 2020), instead of human speech. Specifically, we chose the *Bio* model, which was pre-trained on a masked-prediction task on a total of 142K audio segments (360 hours) of the `animal` label in the AudioSet ontology (ID: `/m/0jbk`) and VGGSound class group. It's architecture is based on HuBERT's *base* model, and contains 12 encoder transformer layers (TL).

SSL pre-trained on human speech: In order to directly compare our performance against AVES-Bio, we select the HuBERT *base* model, pre-trained on a masked-prediction task. In addition, we also look at the *base* WavLM, denoted as WLM, based on its demonstrated effectiveness in animal call and caller classification (Sarkar and Magimai.-Doss, 2023; Sarkar and Magimai.-Doss, 2024; Sarkar et al., 2025), as well as its versatility in speech processing tasks as benchmarked on the SUPERB challenge (S.-w. Yang et al., 2021). Finally, we also use the *base* Wav2Vec2 model, denoted as W2V2, pre-trained on a constrastive task. All three models were pre-trained on the 960-hour Librispeech dataset.

SSL pre-trained and fine-tuned on human speech: For our second study, we assess the impact of fine-tuning on models pre-trained on human speech for bioacoustic tasks. To that end, we

use WLM fine-tuned on 100 hours of Librispeech, and W2V2 fine-tuned on both 100 and 960 hours of Librispeech. All 3 models are fine-tuned on a ASR task[1].

Fusion: We also compute a simple fusion representation as comparison to the other features. For each vocalization segment, we simply compute the mean across the posterior probabilities of all the other features, and then take its argmax.

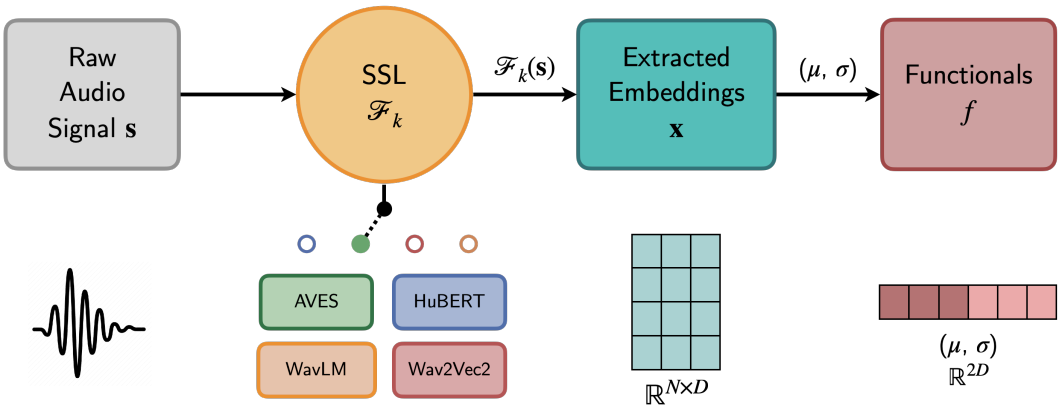


Figure 7.2 – Feature representation extraction pipeline.

The general pipeline for obtaining a feature vector for a given vocalization segment is illustrated in Figure 7.2. We obtain the features from these each of the SSL models $\mathcal{F}$, by first giving them the animal vocalizations $\boldsymbol{s}$ as inputs resampled at 16 kHz. We extract the variable-length embeddings $\boldsymbol{x} \in \mathbb{R}^{N\times D}$ output for each frame. Then, we transform them into fixed-length vocalization-level representations by computing and aggregating first and second order statistics across the temporal axis, resulting in a final feature functional representation $\boldsymbol{f} \in \mathbb{R}^{2D}$. For our work, we extract the embeddings of the CNN and all encoder transformer layers (TL) of $\mathcal{F}$, since we are interested in investigating the features at a layer level.

7.3 Experiments and Analysis

This section looks at the classification performance of the extracted feature representations. In order to compare and evaluate the saliency of the different features, we use same classifier as the Chapter 5 and 6: a simple, non-linear MLP, composed of three blocks of [Linear, LayerNorm, ReLU] layers, with 128, 64, and 32 number of hidden units respectively, followed by a final linear layer.

We train the classifier for 30 epochs using cross-entropy loss, and employ a early-stopping criterion, where training is stopped if no improvement is observed on the *Val* set for 10 consecutive epochs. The optimization consists of Adam, with a η-scheduler of factor 0.1 and

[1]All fine-tuned models are obtained from Huggingface, namely from the `facebook`, `microsoft`, and `patrickvonplaten` repositories.

patience of 10 epochs. We evaluate the performance through Unweighted Average Recall (UAR) as the metric to account for any class imbalance.

7.3.1 Pre-Training Domain Analysis

In this sub-section, we analyze the impact of pre-training domain by comparing AVES against HuBERT. Figure 7.3 shows that HuBERT outperforms AVES in the initial and final layers for IMV. Both models show that the initial transformer layers are more important for this task, indicating that this trend is not specific to speech-based pre-training. The loss of substantial spectral information in these Marmoset calls when down-sampled to 16 kHz likely affects the overall performance (Sarkar and Magimai.-Doss, 2024). For Watkins, we see that AVES's initial layers are not as salient as later ones, where as HuBERT's middle layers are conversely the least useful. In the Abzaliev dataset, AVES performs better overall, with both the initial and later layers contributing comparably. HuBERT, on the other hand, does not scale well, and follows the same downwards trend as IMV. Overall, the results indicate that pre-training on bioacoustic data can provide marginal improvements in some datasets.

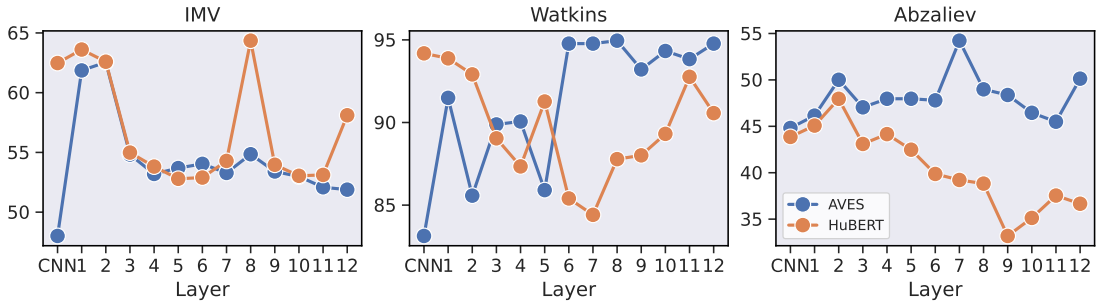


Figure 7.3 – Layer-wise UAR [%] performance of AVES (•) against HuBERT (•).

7.3.2 Fine-Tuning Analysis

Fine-tuning yields mixed effects across both models and datasets, as shown in Figure 7.4. In several cases, we observe that fine-tuned models do not consistently outperform their base counterparts, particularly in W2V2-960h, with performance gains being marginal at best. Notably, fine-tuning on more speech data, such as the 960-hour W2V2, sometimes leads to a decline in performance in later layers, as seen on IMV and Abzaliev. This suggests that fine-tuning on speech may push models to learn task-specific features that don't generalize as well to certain bioacoustic tasks.

Interestingly, for non-fine-tuned models, earlier layers often capture enough general acoustic features to perform adequately. However, for fine-tuned models, selecting the optimal layer becomes more important, as different layers may capture more specialized representations that could benefit certain tasks. This points to the fact that fine-tuning creates more task-

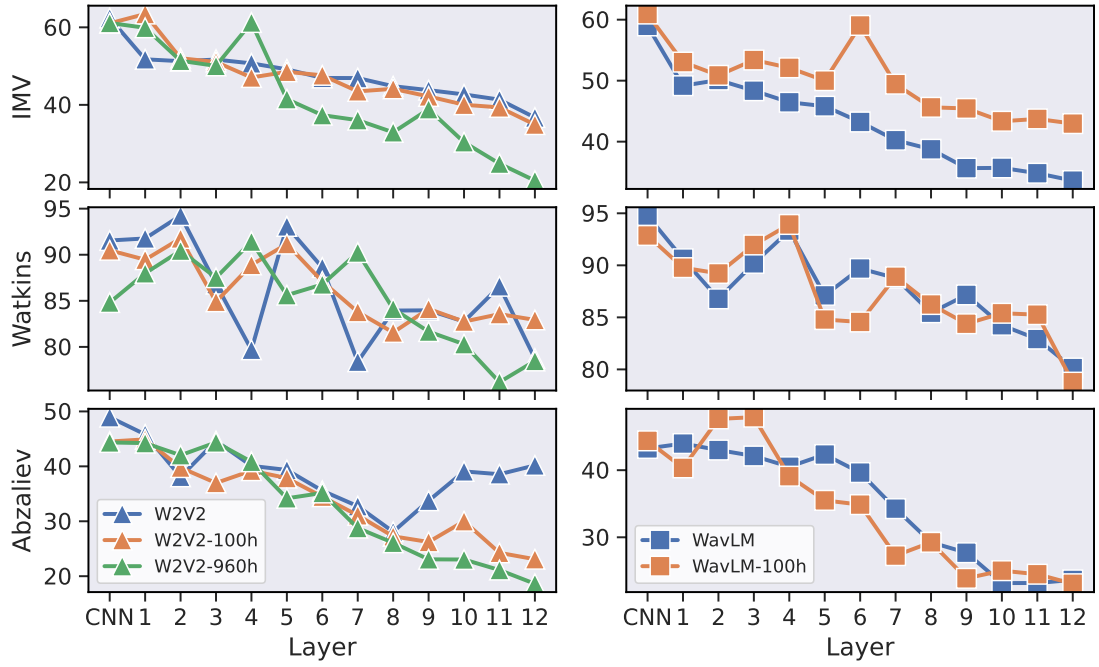


Figure 7.4 – UAR of W2V2 (▲) and WLM (■) against their fine-tuned versions.

specific representations, making careful layer selection more necessary for specific bioacoustic tasks.

7.3.3 Comparative Analysis

Finally, we look at the general classification performance. Table 7.3 tabulates the result of the layers yielding the highest scores from the different features.

Table 7.3 – UAR scores [%] on the best feature layer, on *Test*. Best performance is **bolded**, second best is underlined.

Type	$\mathcal{F}$	IMV	Watkins	Abzaliev
PT	AVES	62.54	**94.95**	**54.23**
	HuBERT	**64.35**	94.18	47.96
	WavLM	58.98	94.78	43.97
	W2V2	62.40	94.25	48.95
PT + FT	WavLM-100h	60.93	93.93	47.90
	W2V2-100h	63.44	91.77	44.91
	W2V2-960h	61.25	91.42	44.36
	Fusion	62.48	94.78	48.95

We can observe that the best scores are from the AVES and HuBERT models, both of which

consist of the same architecture, pre-text task, and loss function. HuBERT and AVES yield very comparable performances for both IMV and Watkins, indicating that HuBERT's representations are robust for call-type classification tasks across different species. AVES achieves a higher score on the Watkins dataset, suggesting that for this specific task, pre-training on bioacoustic data yields a small but notable improvement for species classification. Additionally, we can clearly observe that all the best scores are from the PT category, as well as the second best scores with the marginal exception W2V2-100h on the IMV dataset. This demonstrates that further fine-tuning pre-trained speech models on an ASR task does not consistently bring us any advantage over the pre-trained alone for bioacoustics classification tasks. It suggests that the pre-trained representations may already be optimized, and fine-tuning might not always yield significant benefits. Lastly, we observe that a fusion of all features over their best layers doesn't yield a more salient representation than the best performing model, although it can outperform some of the others.

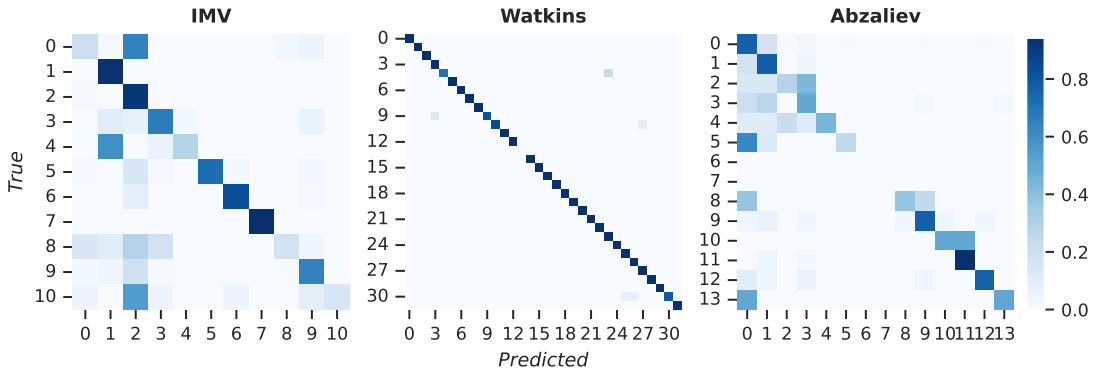


Figure 7.5 – Confusion matrices of the best feature layers' fusion.

Figure 7.5 shows the classifier's performance of the fusion features through confusion matrices. We can observe a good classification alignment for the three datasets. For IMV, there is a noticeable trend of false positives for call-type ID 2, likely due to its high occurrence in the dataset, and wide spectral range, causing an overlap of acoustic features with the other classes. The Watkins dataset is unsurprisingly the easiest to classify, likely because of the clear acoustic and spectral differences in the various species vocalizations, as well as the high variance in segment lengths. Class ID 13 only had two samples which results in an empty row. In the Abzaliev confusion matrix, we observe some confusion between certain call-types, namely the different barks (IDs 0–5) which may contain overlapping acoustic features. Some classes had very few samples (ID 6) or were removed during data preprocessing (ID 7), resulting in empty rows.

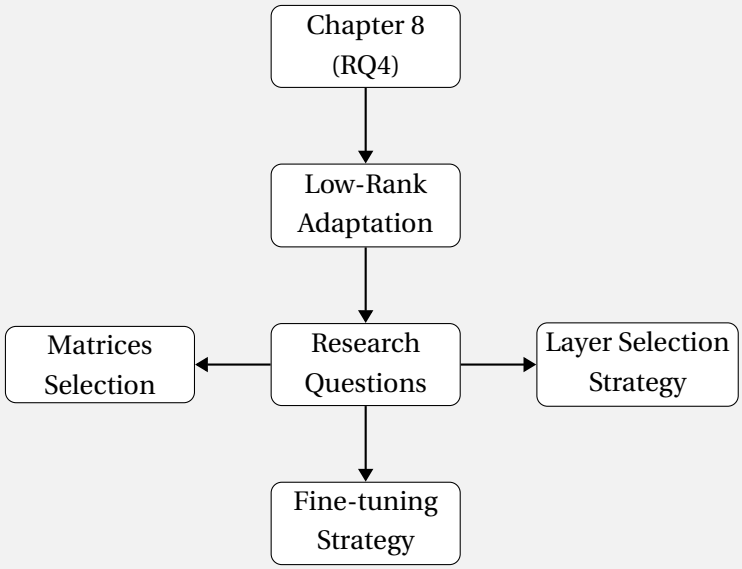
7.4 Conclusions

This chapter presented a comparison of self-supervised learning models pre-trained on human speech and animal vocalizations for bioacoustic tasks. Through two distinct lines of investigation, we first examined the impact of pre-training domains by comparing models pre-trained on human speech and animal vocalizations. The results indicated that pre-training on bioacoustic data mostly yields comparable performance to pre-training on speech, but can offer limited advantages in select contexts. In our second line of investigation, we explored whether fine-tuning pre-trained speech models on ASR could further enhance their ability to capture structured patterns in animal vocalizations. We found that fine-tuning yielded inconsistent results, suggesting that the general-purpose representations learned during pre-training may already be well-suited for bioacoustic tasks, and further fine-tuning on speech does not consistently provide additional benefits.

In conclusion, our results highlight the utility of pre-trained speech models for bioacoustic tasks, even without further fine-tuning. Future work could explore attention mechanisms in SSL models to gain deeper insights into how these models interpret and process specific features of animal vocalizations.

8 Adaptation of Speech and Bioacoustics Models

Chapter Schematic Overview



Publication Note

The material presented in this section is adapted from the following publications:

- E. Sarkar, A. Mohammadi, and M. Magimai-Doss (July 2025). "Adaptation of Speech and Bioacoustics Models". In: Idiap-RR-05-2025.

8.1 Introduction

In Chapter 7, we examined whether fine-tuning models pre-trained on human speech could improve processing of animal vocalizations, but found no consistent gains using publicly available models fine-tuned on ASR. In this chapter, we investigate whether fine-tuning the aforementioned SSL models directly on the downstream bioacoustic data yields better

performance on the same classification tasks.

Fine-tuning a pre-trained model on a downstream task or domain is the second step of the typical SSL framework, as explained in Section 2.4.2. However, in standard fine-tuning, the entire parameter set of the network is updated, which can quickly become exceedingly computationally expensive or even infeasible. The advent of large foundation models has lead to the development of a number of parameter efficient fine-tuning (PEFT) techniques for downstream tasks. The core idea behind PEFT approaches is to only strategically update a small subset of parameters, while keeping the majority frozen, thereby greatly reducing the computational cost and tuning time.

To this end, we adopt Low-Rank Adaptation (LoRA) (Hu et al., 2022) for parameter-efficient fine-tuning (PEFT) and apply it to two architecturally identical models: HuBERT (pre-trained on human speech) and AVES (pre-trained on bioacoustics). We focus exclusively on the call-type identification (CTID) task and conduct systematic ablations to understand the adaptation process. Specifically, we explore which permutation of transformer projection matrices to optimize, which encoder layers permutations to fine-tune, and whether to freeze or drop the remaining layers, in order to achieve better performance. Moreover, having observed a progressive decline in representational quality across deeper layers in the previous chapter, we examine whether this layer-wise trend changes when models are fine-tuned on domain-specific data.

The remainder of this chapter is organized as follows. Section 8.2 provides an overview of parameter efficient fine-tuning and parameter pruning, while Section 8.3 outlines the research questions and experimental methodology for the different experiments. Finally, Section 8.4 present the results from the various studies, and Section 8.5 concludes the chapter.

8.2 Parameter Efficient Fine-Tuning and Parameter Pruning

The following Section 8.2.1 gives a brief overview of Low-Rank Adaptation (LoRA), a modern PEFT adaptation technique which has gained a lot of prominence thanks to its simplicity and effectiveness. We also introduce the notion of parameter pruning in Section 8.2.3.

8.2.1 Low-Rank Adaptation (LoRA)

During training or fine-tuning, a model's parameters are updated through backpropagation, as defined in Equation (2.2). Although these weight parameters $\boldsymbol{w}$ are full-rank matrices, they have been shown to reside in a much lower-dimensional subspace, i.e. to have low 'intrinsic dimension' (Aghajanyan, Gupta, and Zettlemoyer, 2021). Likewise, (Hu et al., 2022) demonstrated that the fine-tuning updates $\Delta\boldsymbol{w}$ themselves exhibit a low 'intrinsic rank'. Consequently, one can efficiently parameterize these updates by decomposing $\Delta\boldsymbol{w}$ into the product of two

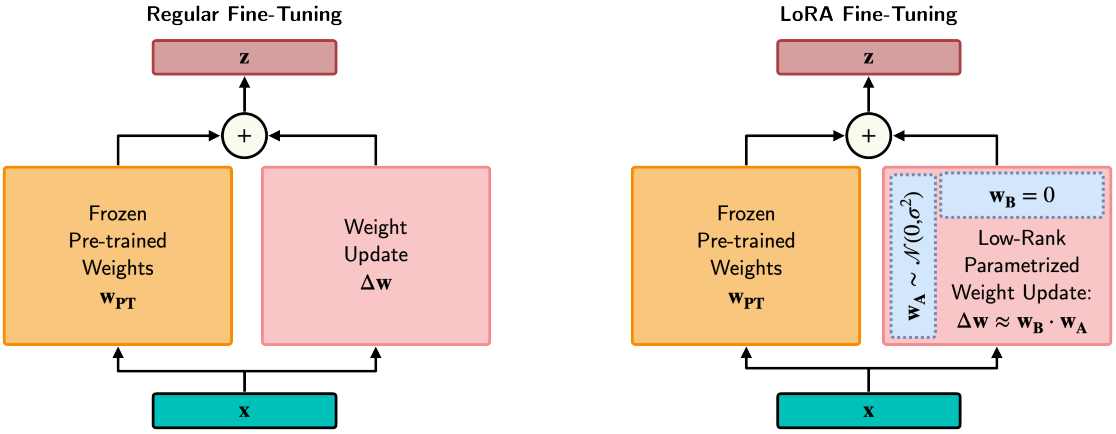


Figure 8.1 – Regular fine-tuning compared to LoRA adaptation. $\boldsymbol{x}$ and $\boldsymbol{z}$ are the input and output.

low-rank matrices $\boldsymbol{w_B}$ and $\boldsymbol{w_A}$. The concept is illustrated in Figure 8.1 and formalized as:

$$\boldsymbol{w}_{FT} = \boldsymbol{w}_{\text{PT}} + \Delta\boldsymbol{w} \tag{8.1}$$

$$= \boldsymbol{w}_{\text{PT}} + \boldsymbol{w_B} \cdot \boldsymbol{w_A} \tag{8.2}$$

where $\boldsymbol{w_A} \in \mathbb{R}^{r\times n}$ and $\boldsymbol{w_B} \in \mathbb{R}^{m\times r}$ for a weight matrix $\boldsymbol{w} \in \mathbb{R}^{m\times n}$. Here, $\boldsymbol{w}_{FT}$ and $\boldsymbol{w}_{PT}$ denote the fine-tuned and pre-trained weights, respectively. We initialize one matrix with random Gaussian values $\boldsymbol{w_A} \sim \mathcal{N}(0,\sigma^2)$, and the other as a zero matrix $\boldsymbol{w_B} = \mathbf{0}$, ensuring that the model's initial output matches the pre-trained model. During the fine-tuning process, both the pre-trained weights $\boldsymbol{w}_{PT}$ and the new *adapters* $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$ are used to compute the hidden states $\boldsymbol{z}$ during the forward pass. However, during the backward pass, only the gradients of low-rank matrices are required to be computed and optimized – the original pre-trained parameters remain frozen. This selective updating drastically reduces the computational cost compared to regular 'full' fine-tuning.

In practice, there are two additional hyperparameters: a constant scaling factor α and the rank $r \ll \min(m,n)$. The modified forward pass, where $\boldsymbol{x}$ is the input and $\boldsymbol{z}$ the output, is thus defined as:

$$\boldsymbol{z} = \boldsymbol{w}_{\text{PT}}\boldsymbol{x} + \frac{\alpha}{r}\boldsymbol{w_B}\boldsymbol{w_A}\boldsymbol{x}, \tag{8.3}$$

Typically LoRA is applied only to the weight matrices in the attention block of transformer-based models during fine-tuning, while the feed-forward module remains unchanged. This approach reduces the number of trainable parameters without compromising the integrity of the pre-trained representations. To the best of our knowledge, LoRA has not yet been employed to transfer models from human speech processing to the bioacoustics domain.

8.2.2 LoRA Adapters in Transformers

Having introduced the main principles of Low-Rank Adaptation, we now consider how these adapters w_A and w_B are integrated into the Transformer architecture shared by HuBERT and AVES. Inserting adapters at appropriate locations allows us to adapt large pre-trained models with minimal parameter updates, while preserving the bulk of the original weights.

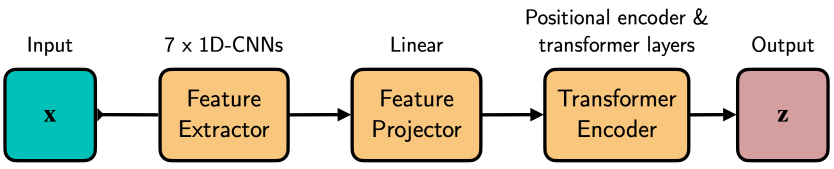


Figure 8.2 – Transformer architecture of HuBERT and AVES.

As shown in Figure 8.2, the Transformer model consists of three main modules that transform raw audio x into context-aware feature representations z:

- **Feature extractor**: seven 1D convolutional layers of different window lengths and shifts, alongside GeLU activation functions and LayerNorms. This block operates directly on the raw waveform, and converts the input audio signal into embeddings of size 512.
- **Feature projector**: a fully-connected layer, preceded by a LayerNorm and followed by a Dropout. This layer projects the output of the feature extractor embeddings from 512 into 768 dimensions.
- **Transformer encoder**: the core of the model, operating on 768-dim vectors, and itself consisting of:
  - One **positional encoder**: convolutional and GeLU layers that inject relative position information.
  - Multiple **Transformer (encoder) layers**: each composed of a *self-attention block* and a two-layer *feed-forward network*. Figure 8.3 illustrates one such layer. Note that the LayerNorm, Dropouts, skip connections, and activations have been omitted for simplicity.

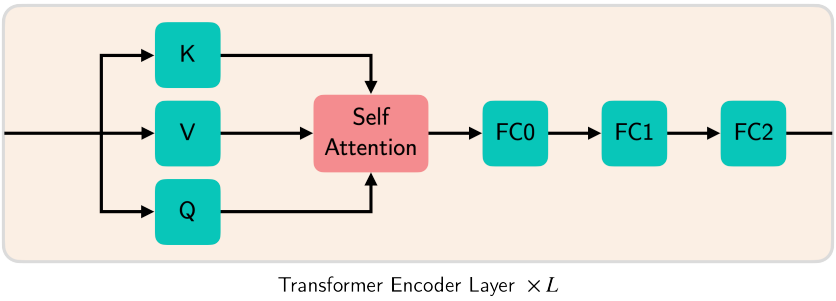


Figure 8.3 – Simplified transformer encoder layer.

Within each Transformer encoder layer, there are multiple candidate weight matrices where LoRA adapters can be inserted to capture task-specific adjustments. In the diagram:

- The *self-attention* consists of the keys, queries, values (K, Q, V) matrices, as well as the output linear layer, often referred to as O, but here denoted as FC0. The self-attention computation is described in Equation (2.8) and (2.9).
- The *feed-forward network* consists of two fully-connected layers, henceforth referred to as FC1 and FC2.

LoRA adapters may be added to any of these projection matrices (Q, K, V, FC0, FC1, or FC2), enabling the model to learn low-rank updates at these points while keeping all other parameters fixed. By selecting different combinations of adapter placements, one can tailor the fine-tuning process to balance between parameter efficiency and adaptation flexibility.

8.2.3 Parameter Pruning and Layer Dropping

Large pre-trained speech foundation models can be over-parametrized for downstream tasks. Prior work in the literature has shown that structured or adaptive parameter pruning can reduce model size while preserving strong classification performance (Peng et al., 2023). Rather than individual weights, layer dropping, i.e. removing entire layers, has also been investigated as a parameter pruning technique. Numerous layer-dropping strategies such as top-down, bottom-up, and alternating layer removals have been explored in transformer-based models, achieving up to 40% reduction in model size with only a 2% drop in downstream accuracy (Sajjad et al., 2023). Although these approaches have proven effective in NLP, their application to bioacoustics domain and effectiveness in cross-domain adaptation remains unexplored.

8.3 Research Questions and Experimental Methodology

This section formalizes the central research questions guiding our investigation into adapting pre-trained speech and bioacoustic models via LoRA, and defines the experimental design used to answer them.

8.3.1 Encoder Matrix Selection

Based on the possible adapter insertion points identified in Section 8.2.2, we first explore which combinations of weight matrices within the Transformer layers yield the greatest downstream classification performance when fine-tuned with LoRA. We also examine whether extending LoRA fine-tuning beyond the Transformer encoder, specifically to the feature extractor and feature projector, also leads to further improvements. To that end, we formulate the following two research questions:

Q1. Which subset or permutation of transformer projection matrices (K, Q, V, FC0, FC1, FC2) is most effective for LoRA-based fine-tuning?

To answer this, we compare the following adapter configurations:

- [FC1, FC2]: the two-layer feed-forward network only.
- [Q, K]: the self-attention query and key projections.
- [Q, K, V]: all three self-attention projections.
- [Q, K, V, FC0]: self-attention as well as the attention output projection.
- [Q, K, V, FC0, FC1, FC2]: all self-attention and feed-forward projections.

We individually fine-tune a pre-trained HuBERT under each of these five different settings on the *Train* set, and measure UAR on *Test*, thereby identifying which permutation delivers the best downstream adaptation.

Q2. Does applying LoRA adapters to the feature extraction and/or feature projection modules, in addition to the Transformer encoder, improve classification performance?

Although parameter-efficient fine-tuning typically focuses on the Transformer layers alone, strong acoustic domain shifts, such as moving from human speech to non-human animal vocalizations, may potentially benefit from adapting earlier, pre-encoder network components. To investigate this, we compare three configurations:

- *Encoder only*: LoRA adapters inserted in the Transformer encoder layers only (baseline). We insert the adapters on the optimal matrix permutation found from Q1.
- *Projector* + *encoder*: adapters applied to both the Transformer encoder and the feature projection fully-connected layer.
- *Extractor* + *projector* + *encoder*: LoRA adapters are applied to the Transformer encoder and the feature projection layer, while the feature extractor block is fully fine-tuned, instead of through LoRA decomposition. This is due to a limitation in the implementation of the PEFT HuggingFace library, which currently supports LoRA only on linear modules. However, we keep the feature extractor fully trainable, such that its convolutional filters can still directly adapt to the specific characteristics of bioacoustic signals.

We hypothesize that including the feature projector, a simple affine mapping, will yield additional gains, while adapting the convolutional extractor may have uncertain effects, given its role in low-level signal processing and the risk of disrupting learned acoustic filters. Moreover, the impact of full fine-tuning, as opposed to LoRA-based adaptation, may differ significantly in these components.

By systematically evaluating these configurations on our CTID task with UAR, we will identify which adapter placement strategy offers the best balance of parameter efficiency and performance.

8.3.2 Layer Selection Strategies

Choosing which encoder layers to adapt is an important decision in parameter-efficient fine-tuning. Rather than fine-tuning *all* layers, we investigate whether updating only a particular subset can yield comparable or better performance, and whether there exists a systematic strategy for selecting these layers. Prior work and previous chapters have shown that initial layers in speech SSL models work much better than the later layers for bioacoustics tasks. We therefore ask:

Q3. Which layer selection strategy for the Transformer encoder yields the most effective performance after fine-tuning ?

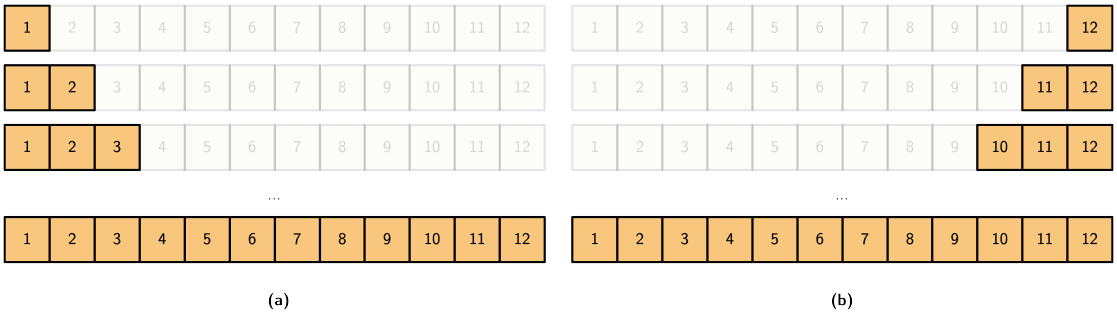


Figure 8.4 – Layer selection strategies: (a) bottoms-up. (b) top-down. The numbers corresponds to transformer encoder layers. Each row represents a different layer permutation, eg. 1, 1–2, 1–3, etc.

To answer this question, we employ two different layer selection strategies, as shown on Figure 8.4. For each strategy, we compute all permutations:

- *Bottoms-up* strategy incrementally adapts the encoder from its lowest layer upwards. In one permutation, we only use the output embeddings of the first layer, then in the second combination, we use the ones of the second layer, with the input having gone through the first two layers, and so on till the final permutation where the input traverses all the layers and we use the output embeddings of the final layer.

  For L total encoder layers, denoted as $l_1, l_2, \ldots, l_L$, we define an independent fine-tuning configuration for each $k \in \{1, 2, \ldots, L\}$:

  $$\mathcal{A}_k = \{l_1, l_2, \ldots, l_k\}, \qquad \mathcal{F}_k = \{l_{k+1}, \ldots, l_L\},$$

  where $\mathcal{A}_k$ denotes the set of adapted layers with LoRA adapters, and $\mathcal{F}_k$ the set of frozen layers. We then measure downstream classification performance for each k, thereby quantifying the incremental contribution of the first k layers to the adaptation process.

  Since the later layers typically learn more task-specific information, we hypothesize that fine-tuning the lower layers could still bring substantial improvements, as these typically encode more acoustic information.

- *Top-down* strategy conversely starts by first adapting the highest-level layer embeddings only, and progressively includes lower layers. In this case, we define our configurations as:

$$\mathcal{A}'_k = \{l_{L-k+1}, l_{L-k+2}, \ldots, l_L\}, \quad \mathcal{F}'_k = \{l_1, \ldots, l_{L-k}\},$$

  where $\mathcal{A}'_k$ are the layers adapted with LoRA, and $\mathcal{F}'_k$ are frozen. By evaluating classification performance for each k, we assess how the inclusion of progressively lower-level layers impacts adaptation.

  In this strategy, it could be argued that starting adaptation with the top layers could accelerate the domain adaptation and force the model to learn representations more relevant to the animal-specific vocalizations.

8.3.3 Fine-Tuning Strategies: Probing, Freezing, and Pruning

Rather than simply freezing unselected layers during LoRA adaptation, parameter-pruning research detailed in Section 8.2.3 suggests that removing those layers from the model entirely may further improve efficiency without degrading performance. We therefore compare three distinct adaptation strategies, and formulate our question as:

Q4. Which approach yields the best downstream performance between (a) simple linear probing, (b) LoRA fine-tuning with layer freezing, and (c) LoRA fine-tuning with layer pruning ?

The three scenarios are illustrated in Figure 8.5, and explained below:

(a) *Linear probing*: All encoder layers remain frozen. We simply extract the output embedding of the selected layer(s), apply mean-pooling, and train a single linear classifier on top. Note that this scenario is essentially identical to the one used in Chapter 7, with the key difference that we only employ a single linear classifier instead of an MLP. Using the same classifier across all adaptation scenarios ensures a fair comparison.

(b) *LoRA + freezing*: LoRA adapters are inserted into the selected layers and fine-tuned, while all other layers remain frozen and only participate in the forward pass.

(c) *LoRA + pruning*: Selected layers receive LoRA adapters and are fine-tuned, but all other layers are removed from the model, and the classifier is applied directly on the output of the highest adapted layer.

In all scenarios, we apply the LoRA adapters to the optimal matrix permutation found from Q1. By evaluating each strategy on both HuBERT and AVES, we can determine whether dropping unused layers offers any advantage over freezing them, and how both compare to a classic linear-probing baseline. Finally, to assess which strategy performs best, we conduct all experiments on both the Abzaliev and IMV datasets.

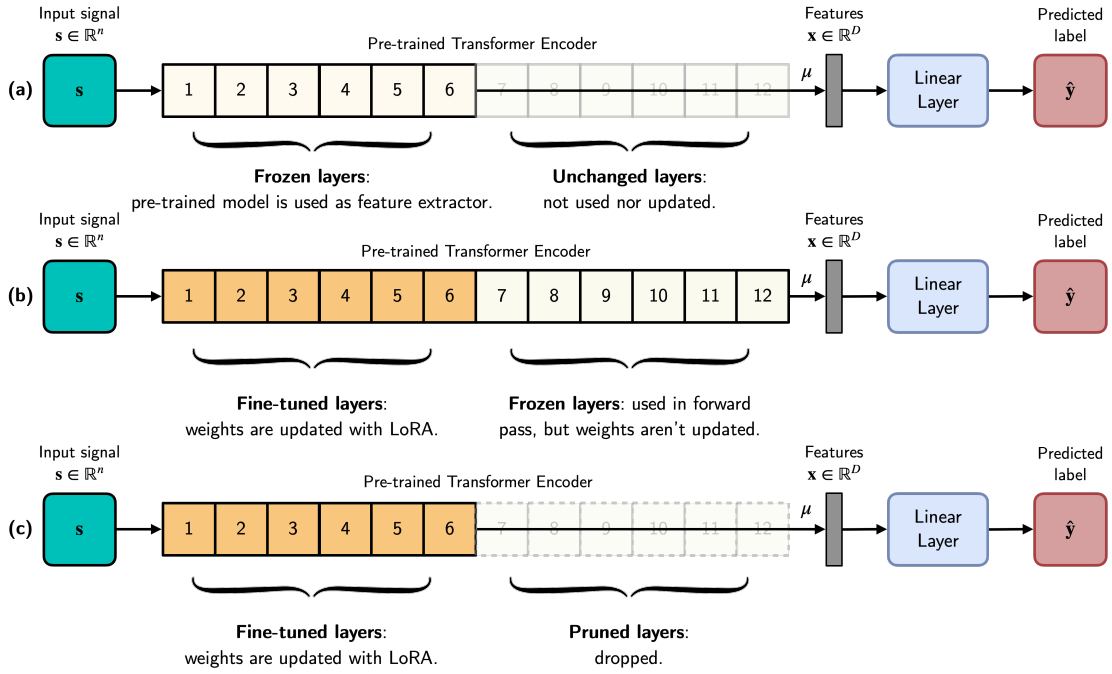


Figure 8.5 – Three evaluation scenarios of a pre-trained SSL model using a linear classifier. This example depicts the case where layers 1–6 are selected and used for classification, while any remaining layers are either ignored, kept frozen, or pruned, depending on the scenario. **a) Linear probing**: all layers of the pre-trained model are frozen. The input signal $\boldsymbol{s}$ passes through the layers 1–6. The output embedding from layer 6 is extracted and given to a linear classifier, which is trained. The remaining layers are ignored. **b) LoRA fine-tuning with freezing**: LoRA adapters are inserted into the selected layers 1–6, which are adapted, while the others 7–12 remain frozen. **c) LoRA fine-tuning with layer pruning**: the model is pruned such that only the selected layers 1–6 are retained and then fine-tuned using LoRA, while all the others are removed from the model entirely. Note that layers 7–12 are functionally identical in scenarios (a) and (c): they are unused in both cases. We distinguish them visually to emphasize that in (c) they are explicitly removed from the model, whereas in (a) they are simply ignored. In each case, the output embeddings of the pre-trained model are mean-pooled over the temporal axis to produce a single functional feature vector $\boldsymbol{x}$. In practice, a LayerNorm layer is also implemented before the linear layer for robustness.

8.4 Results and Analysis

8.4.1 Hyperparameter Selection

Given the large number of fine-tuning configurations and model permutations, we performed a preliminary grid search on HuBERT to identify a single set of LoRA hyperparameters that could then be kept constant across all subsequent experiments. The search spanned learning rate η, rank r, scaling α, dropout, weight decay, and number of epochs.

To ensure these settings generalize across different adaptation structures, we ran the search independently for each of the five matrix permutations defined in Q1, and for each of the twelve bottoms-up layer selections while keeping the unselected layers frozen. In total, this amounted to approximately 900 trials on the CTID task. The hyperparamters optimized in the

grid search are given in Table 8.1.

Table 8.1 – Search space to find optimal hyperparameters.

Hyperparameters	Search Space	Optimal Value
α	[1, 2, ..., 60]	3
r	[4, 8, ..., 64]	60
Dropout	[0, 0.1, 0.2, ..., 1.0]	0.3
η	1e[-3, -2, -1]	1e-3
Weight decay	1e-9 – 9.67e-2	8e-09
Max. epochs	[1, 2, 3, 4, 5]	5

We found that a low learning rate (10^{-3}), a high adapter rank ($r = 60$), and a moderate scaling factor ($\alpha = 3$) produced consistently strong performance, with optimal dropout of 0.3 and minimal weight decay ($8 \cdot 10^{-9}$). These settings balance adaptation capacity against overfitting risk and are used throughout the rest of our studies.

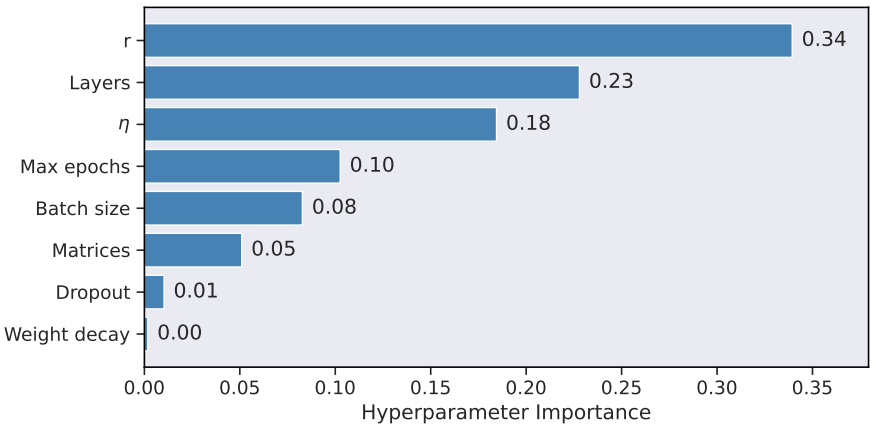


Figure 8.6 – Hyperparameter importance on HuBERT, as estimated by the fANOVA algorithm.

The hyperparameter importance plot in Figure 8.6 quantifies each parameter's contribution to the variation in downstream UAR, as estimated by the fANOVA algorithm (Hutter, Hoos, and Leyton-Brown, 2014). The results indicate that adapter rank r is by far the most influential hyperparameter, reflecting the fact that increasing the latent dimensionality of the LoRA update substantially enhances the model's adaptation capacity. Next in importance is layer selection, confirming that the choice of encoder layers that receive adapters does affect the performance. The learning rate η remains critical, consistent with its central role in gradient-based optimization, but ranks below rank and layer decisions. The number of epochs and batch size exhibit moderate impact, suggesting that training duration and mini-batch stability provide incremental gains once rank, layers, and η are set. The choice of projection matrices, formulated in Q1, seems to have only a modest effect once the principal LoRA capacity and layer locations are determined. Finally, LoRA dropout and weight decay show near-zero importance, implying that explicit regularization is largely unnecessary under LoRA fine-

tuning for CTID.

8.4.2 Matrix Selection Results (Q1)

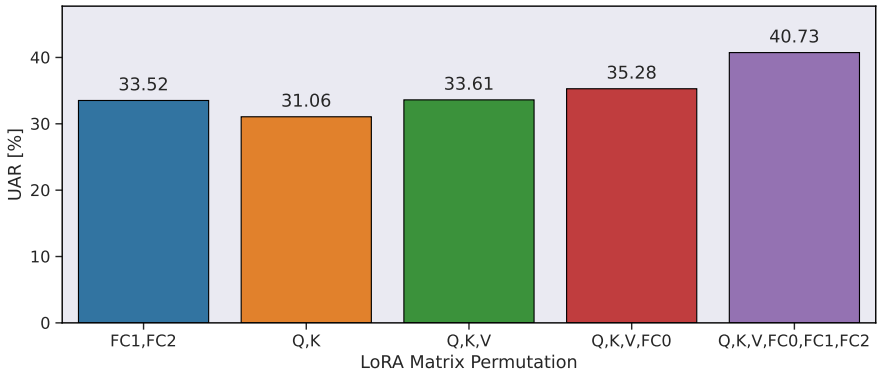


Figure 8.7 – Best UAR [%] for each LoRA adapter configuration on layers 1–12. Fine-tuning all matrices yields the best performance.

Figure 8.7 shows the highest UAR score achieved for each of the five different LoRA adapter matrix configurations defined in Q1. To ensure a fair comparison across matrix combinations, we fix the selected layers to HuBERT encoder layers 1–12 for all experiments. All results are obtained on the Abzaliev dataset for the call-type classification (CTID) task. For each configuration, we report the best UAR achieved across our full hyperparameter sweep. The results exhibit a clear, monotonic progression:

$$\text{Q, K} < \text{Q, K, V} < \text{Q, K, V, FC0} < \text{Q, K, V, FC0, FC1, FC2.}$$

In other words, performance steadily increases as more projection modules are adapted. Fine-tuning only the query and key projections yields the lowest UAR, with each successive addition (value, attention output, feedforward layers) leading to higher scores. This progression highlights that granting the model greater adaptation capacity, by increasing the number of LoRA-enabled projections, consistently improves downstream accuracy, with the full set of adapters delivering the best result.

8.4.3 Layer Selection Strategy Results (Q2 & Q3)

Based on the previous results, we fix the matrix permutation to include all the aforementioned matrices, and now aim to identify which layer selection strategy and permutation yields the best fine-tuning results.

Figure 8.8 compares the best UAR scores across different layer selection strategies for both AVES and HuBERT. We observe that in both cases, fine-tuning the feature extraction (FE) layers severely degrades performance. Fine-tuning the feature projection (FP) alone does not significantly improve performance relative to other strategies, but it also does not degrade

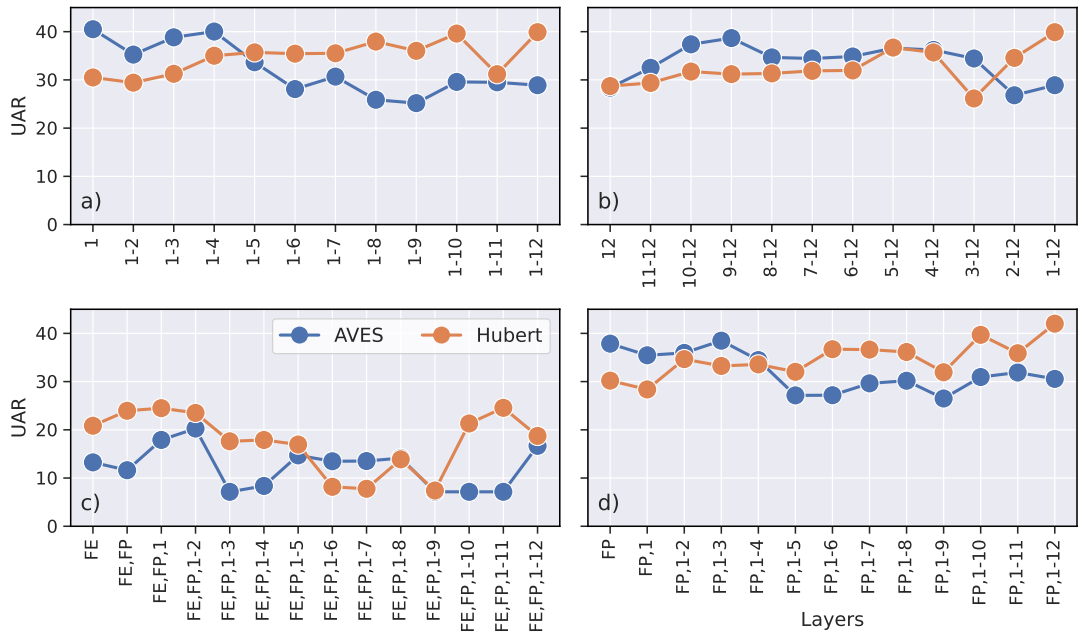


Figure 8.8 – Layer selection strategy UAR [%] results: (a) bottoms-up, (b) top-down, (c) FE + FP + bottoms-up, (d) FP + bottoms-up.

it, suggesting that FP adaptation is optional rather than essential. Furthermore, bottoms-up and top-down layer selection strategies yield comparable results, generally achieving scores in the range of 30–40% bracket for all layer permutations. Finally, neither AVES nor HuBERT consistently outperforms the other across all layer selections. However, HuBERT appears to perform slightly better in the later layers in the bottoms-up strategy, with or without feature projection tuning.

8.4.4 Fine-Tuning Strategy Selection (Q4)

In this final research question, we evaluate three paradigms aforementioned in Q4, namely linear probing, LoRA with layer freezing, and LoRA with layer pruning, applied to the Transformer encoder in a bottoms-up layer selection. For fairness, we keep the feature extraction (FE) and feature projection (FP) modules unchanged, since standalone fine-tuning on these sub-modules did not yield consistent gains. We run these experiments on both the Abzaliev and IMV datasets, using AVES and HuBERT feature representations.

Figure 8.9 displays the per-layer UAR performance for each strategy. On the IMV dataset, LoRA fine-tuning, whether with freezing or pruning, consistently and significantly improves performance over simple linear probing across nearly all layers when using AVES, and shows clear gains in the later layers of HuBERT. By contrast, on the smaller Abzaliev dataset, simple linear probing almost always exceeds either LoRA performance, suggesting that LoRA tuning offers limited benefit in low-data scenarios. However, this performance gap on Abzaliev is

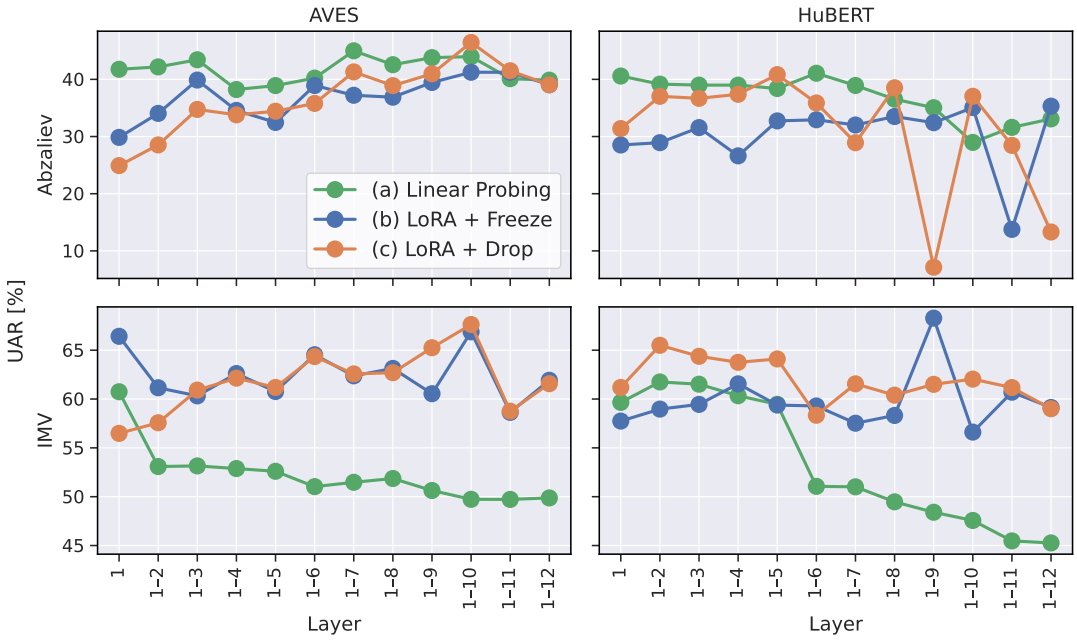


Figure 8.9 – Layer-wise UAR [%] performance of scenarios (a), (b), and (c).

modest compared to the substantial gains that LoRA fine-tuning delivers on the larger IMV dataset. This suggests that LoRA's advantages scale with dataset size, whereas in lower-data scenarios a simple linear probe may be more a reliable choice.

We can also observe that in the case of AVES, both LoRA-tuned models display a general *upward* trajectory for IMV and Abzaliev, whereas the linear probe continues to follow the same downward trend when going deeper in the layers, as seen in previous chapters. This demonstrates that deeper transformer layers in AVES encode increasingly abstract features that can effectively classify calls, but only when these layers are fine-tuned. A linear probe, which freezes the backbone, cannot leverage these deeper embeddings, and thus its performance declines in later layers. In contrast, LoRA injects a small number of trainable parameters into each layer, providing just enough task-specific flexibility to enable each additional layer to contribute positively, yielding a steady upward trend in performance. Practically, this implies that when extracting features from deeper layers within the transformer, one should pair them with parameter-efficient fine-tuning methods, such as LoRA, rather than relying on a fixed feature extractor alone.

8.4.5 Classifier Comparison: Linear Layer vs. MLP

The results obtained in the previous Section 8.4.4 can be directly compared, on the same datasets and feature representations, with those from Chapter 7's Section 7.3.1. In this chapter, we fine-tuned models using LoRA with a single linear output layer, as depicted in Figure 8.5, and compared them to a linear layer baseline. However, in previous chapters, we employed a

MLP, composed of three blocks of [Linear, LayerNorm, ReLU] layers and a final linear layer, to evaluate various feature representations.

Figure 8.10 shows the highest scores of each scenario (a–c) from Figure 8.9, across all layers, alongside the corresponding MLP results from earlier chapters. This allows us to assess the potential benefit of classifier complexity, specifically, to see whether using a non-linear MLP really leads to better performance than a single linear layer.

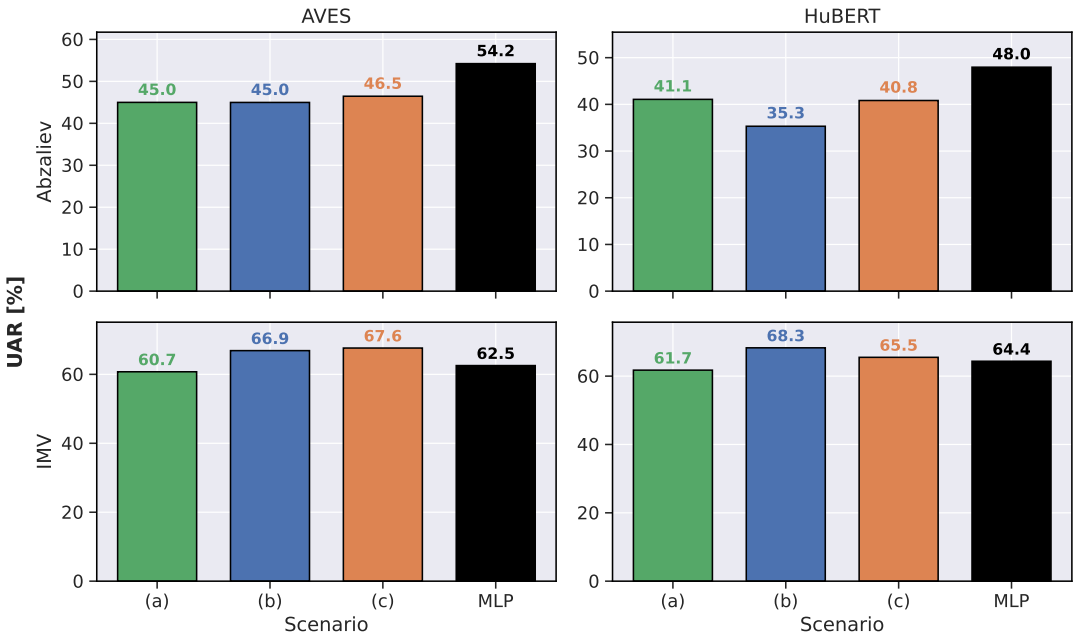


Figure 8.10 – Best UAR results across layers for the (a), (b), and (c), scenarios defined in RQ4, using a linear layer classifier, compared to an MLP classifier.

We can observe that for the Abzaliev dataset, the MLP classifier clearly outperforms the single-layer LoRA variants, (b) and (c), for both AVES and HuBERT, suggesting that the added classifier capacity and non-linearity does help for CTID. However, for IMV, the opposite holds true: both single-layer LoRA models yield higher scores than the MLP classifier, indicating dataset-specific behavior.

Overall, these results do not allow us to draw general conclusions. While increased capacity may help in some cases, it may not be universally beneficial. Further investigation, such as fine-tuning a LoRA model with a non-linear MLP classifier, could give deeper insight into the impact of classifier capacity and non-linearity in this context.

8.5 Conclusions

In this chapter, we studied the potential of parameter-efficient fine-tuning (PEFT) for adapting large speech and bioacoustic SSLs models. We showed that Low-Rank Adaptation (LoRA) can

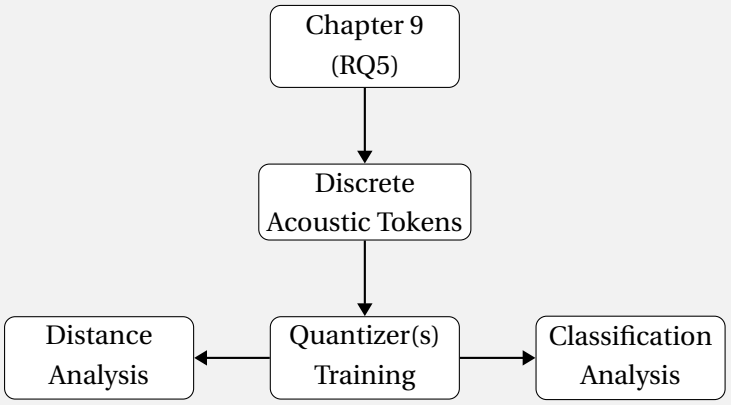
greatly enhance call-type classification of animal vocalizations when sufficient labeled data is available. We systematically investigated a number of research directions by conducting a series of controlled experiments regarding LoRA adapter placements, layer selections, and fine-tuning strategies, and arrived at the following insights:

- Transformer encoder matrix selection: adapting an increasing subset of projection matrices yields steadily higher performance, with adaptation of entire self-attention and feed-forward projections achieving the best UAR.
- LoRA adapters scope: extending LoRA adapters beyond the Transformer encoder to the feature projection layer yields only marginal gains, whereas fine-tuning the convolutional feature extractor consistently and significantly degrades downstream performance.
- Layer selection strategy: neither 'bottoms-up' nor 'top-down' layer selection strategies clearly outperforms one another. Both produce comparable results when adapters are placed on the same matrices.
- Fine-tuning strategy: on the larger IMV dataset, LoRA fine-tuning (with either freezing or pruning) substantially outperforms simple linear probing across nearly all layers. In contrast, on the smaller Abzaliev dataset, simple linear probing remained more reliable, though the performance gap was modest. This indicates that LoRA's efficacy scale with dataset size.
- Classifier selection: LoRA adaptation with a single linear layer outperforms a deeper 4-layer MLP classifier head on IMV, while the reverse is seen for Abzaliev, indicating further investigation is needed to draw firm conclusions.

In conclusion, the overall results indicate that low-rank adaptation is a highly effective PEFT method and powerful tool for bioacoustic classification when ample data is available, enabling even deep transformer layers to contribute meaningfully. In low-data settings, however, a classic linear probe may still be preferable.

9 Leveraging Sequential Structure in Animal Vocalizations

Chapter Schematic Overview



Publication Note

The material presented in this section is adapted from the following publications:

- E. Sarkar and M. Magimai-Doss (July 2025a). "Leveraging Sequential Structure in Animal Vocalizations". In: Idiap-RR-06-2025.

9.1 Introduction

In all the previous chapters, we averaged each data sample's extracted feature embeddings $\boldsymbol{x} \in \mathbb{R}^{N \times D}$, into a vocalization-level representations, denoted as functional vectors $f_\mu = \mu(\boldsymbol{x}) \in \mathbb{R}^D$ or $\boldsymbol{f}_{\mu\sigma} = \big[\mu(\boldsymbol{x}), \sigma(\boldsymbol{x})\big] \in \mathbb{R}^{2D}$. While these 'stats-pooled' representations have proven very valuable for classification tasks, bandwidth analysis, and model adaptation, they ignore the sequential aspect of animal calls: each vocalization is treated like an unordered bag of frame-level feature embeddings. This completely overlooks the fact that many animal arrange acoustically distinct sub-vocalization units in a specifically ordered sequences that carry important communicative and syntactic information (Kershenbaum, D. Blumstein, et al.,

2016). The goal of this final chapter is thus to investigate alternate feature representations that can capture the sequential structure within animal vocalizations, and leverage the unutilized temporal information to improve classification performance.

In order to effectively model sub-vocalization unit level sounds, we turn to symbolic speech tokenization. Recent work has shown that discrete audio tokens obtained through vector-quantization of 'continuous' SSL feature embeddings can effectively encode acoustic information, and thus be utilized for many speech and audio tasks (Guo et al., 2025). Based on this prior, we extend this framework to bioacoustics, and explore whether token sequences can also reveal meaningful structure in animal vocalizations and help distinguish call-types or individual callers. A successful framework could even yield an inventory of recurring acoustic sub-vocalization units in animal communication. To the best of our knowledge, this is the first work to explore discrete audio tokens for computational bioacoustics. To that end, we investigate vector quantization (VQ) and gumbel-softmax vector quantization (GVQ) as tokenization methods for capturing the sequential structure in non-human animal vocalizations.

The rest of the chapter is structured as follows. First, Section 9.2 provides a brief overview of sequences in animal vocalizations. Then, Section 9.3 presents an in-depth review of representation learning using discrete audio tokens. Section 9.4 describes our experimental setup, namely the quantizer training protocol, token sequence generation, and post-processing. In Section 9.5, we conduct the pairwise distance analysis, and in Section 9.6 we benchmark the downstream classification performance. Finally, we conclude with implications and directions for future research in Section 9.7.

9.2 Sequences in Animal Vocalizations

The communicative power of sequences in animal vocalizations is well-documented across species, with vocal sequences often serving key biological roles such as territory defense, mate attraction, social bonding, and alarm signaling (Kershenbaum, D. T. Blumstein, et al., 2016). The complexity of these sequences manifests through distinct patterns of acoustic units that are combined in species-specific ways, following implicit or explicit syntactic rules. For instance, songbirds produce vocalizations composed of repeated motifs and notes arranged in recognizable patterns (Catchpole and Slater, 2003), while cetaceans exhibit intricate, temporally-structured acoustic sequences associated with social interaction and individual identification (Mercado and Handel, 2012). Thus, capturing and analyzing the inherent sequential structure in animal vocalizations could substantially enhance our understanding of their communicative function and biological significance.

Several approaches have been proposed in the biological literature to analyze the temporal and structural complexity of vocalizations. These include methods derived from information theory and Markovian analyses of transitions between acoustic units (McCowan, Hanser, and Doyle, 1999), as well as pattern recognition techniques applied directly to acoustic sequences (Kershenbaum et al., 2012). However, these biologically-driven studies often rely

heavily on manual annotations or simple acoustic measurements, limiting their scalability and computational generality.

9.3 Discrete Audio Tokens-based Representation Learning

Many self-supervised speech SSL models, including those we have utilized throughout this thesis, employ discrete token representations during their pre-training stages. Typically, these discrete tokens are derived using a quantization process, either through integrated Vector Quantization (VQ) layers (Baevski, Schneider, and Auli, 2020; Baevski et al., 2020) or offline clustering mechanisms applied to continuous embeddings (W.-N. Hsu et al., 2021). However, such discrete representations are primarily intended to facilitate self-supervised learning objectives, such as masked prediction or contrastive learning, and are usually not directly exposed or utilized during inference or downstream tasks.

In this chapter, we explicitly leverage the discrete tokenization methodology. To do so, we first extract window-level embeddings from a pre-trained SSL model, consistent with our earlier experiments, and subsequently train a separate quantization module which maps the embeddings into sequences of discrete tokens. Note that the quantization is performed independently per frame, thereby preserving the temporal order of the original acoustic events within the vocalization, and is trained separately on extracted embeddings from the pre-trained encoder, using the bioacoustic data of interest. This allows the codebook vectors to adapt specifically to the acoustic characteristics and distributions of the vocalizations being studied.

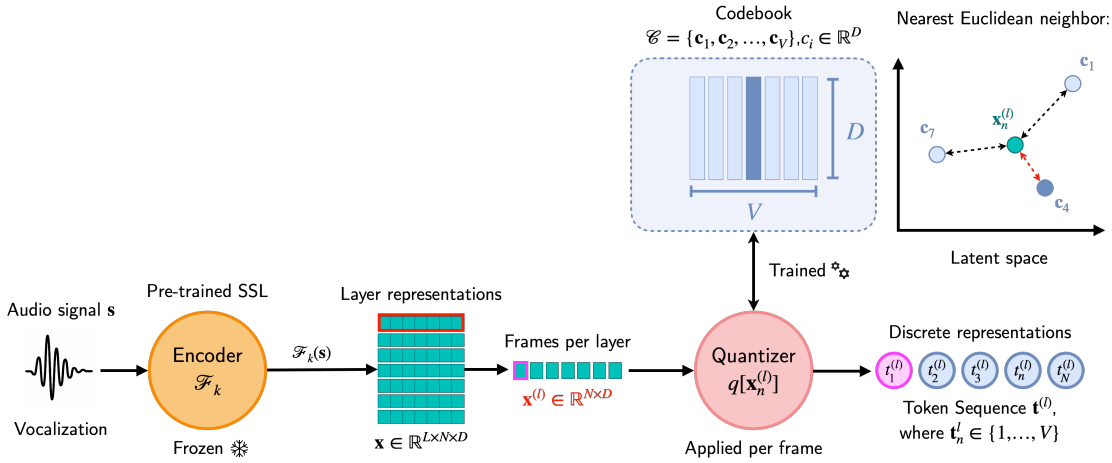


Figure 9.1 – Discrete call tokenization pipeline using vector quantization.

The overall call tokenization pipeline, employed in this work, using vector quantization is illustrated in Figure 9.1. Specifically, a raw audio waveform $\boldsymbol{s}$ is first passed through a pre-trained encoder $\mathcal{F}$, producing continuous layer embeddings $\boldsymbol{x} \in \mathbb{R}^{L \times N \times D}$, where L is the number of layers, N the number of frames in each layer, and D the dimension of each frame. Let

$\boldsymbol{x}_n^{(l)} \in \mathbb{R}^D$ denote the embedding extracted from encoder layer l at frame position n. Each layer embedding is then quantized individually per-frame by a quantization function q, resulting in discrete tokens $\boldsymbol{t}_n^{(l)} = q[\boldsymbol{x}_n^{(l)}]$. Formally, the quantization function maps each embedding from continuous D-dimensional space to a discrete integer token index $q : \mathbb{R}^D \to \{1, 2, \ldots, V\}$ where V denotes the vocabulary size, i.e., the number of unique discrete tokens. Each token index corresponds directly to an entry in a finite set, referred to as the codebook $\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_V\}$, where each code-vector $\boldsymbol{c}_i \in \mathbb{R}^D$ corresponds to the i-th discrete token in the original embedding space. This discretization step effectively compresses the representation since encoding tokens only requires $\lceil \log_2 V \rceil$ bits per frame.

Detailed descriptions of vector quantization and Gumbel-Softmax vector quantization, which are the specific methods employed to train these quantization modules, are provided in Section 9.3.1 and Section 9.3.2, respectively. We specifically leverage them due to their proven effectiveness in quantizing audio embeddings. Note that these are both examples of single-codebook quantizers. Most modern acoustic tokenizers have multiple quantizers. However, for simplicity and clarity, we focus on hand-coded single-codebook ones in this work.

9.3.1 Vector Quantization (VQ)

While traditional clustering methods operate independently of model training, vector quantization integrates a discrete, learnable codebook directly into the neural network (Den Oord, Vinyals, and Kavukcuoglu, 2017), enabling end-to-end optimization via gradient propagation through the quantization step.

We maintain a learnable codebook $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_V\} \in \mathbb{R}^{V \times D}$ of $V = 50$ code-vectors, each of dimension $D = 768$. Given an input embedding $\boldsymbol{x}_n^{(l)}$, the quantization process selects the nearest codebook vector $\boldsymbol{c}_i$ by simply minimizing the Euclidean distance between the two:

$$q[\boldsymbol{x}_n^{(l)}] = \underset{i \in \{1,2,\ldots,V\}}{\operatorname{argmin}} \|\boldsymbol{x}_n^{(l)} - \boldsymbol{c}_i\|_2^2 \tag{9.1}$$

which returns the token index which is the input's discrete token. The codebook vector itself, which we denote as $\boldsymbol{c}_k \triangleq \boldsymbol{c}_{q(\boldsymbol{x})}$, is passed on to subsequent networks.

To allow backpropagation through the non-differentiable nearest-neighbor argmin lookup given in 9.1, a *straight-through estimator* (STE) (Bengio, Léonard, and Courville, 2013) is employed to graft gradients from the quantized output $\boldsymbol{c}_k$ back to $\boldsymbol{x}_n^{(l)}$ during the backward pass. The encoder thus receives learning signals from downstream losses, while the codebook vectors themselves are updated via the VQ loss below. In our case, since we have pre-extracted embeddings, no encoder is updated, and the downstream losses encourage the extracted representations to align with their assigned code-vectors, even though only the codebook parameters are updated. During training, we optimize the VQ loss which is jointly defined as the sum of the codebook and commitment losses:

$$\mathcal{L}_{\text{VQ}} = \underbrace{\|\text{sg}[\boldsymbol{x}_n^{(l)}] - \boldsymbol{c}_k\|_2^2}_{\text{Codebook Loss}} + \underbrace{\beta \|\boldsymbol{x}_n^{(l)} - \text{sg}[\boldsymbol{c}_k]\|_2^2}_{\text{Commitment Loss}}. \tag{9.2}$$

where sg[·] denotes the stop-gradient operator and the beta coefficient is typically set to $\beta = 0.25$. The codebook loss shifts the selected code-vector $\boldsymbol{c}_k$ toward its corresponding input embedding $\boldsymbol{x}_n^{(l)}$, whereas the commitment loss conversely encourages the embedding to move closer to its matched codeword. We iterate $\mathcal{L}_{\text{VQ}}$ over all the layers L and frames N to compute the total cost. While one can also update the codebook via an exponential-moving-average (EMA) scheme (Den Oord, Vinyals, and Kavukcuoglu, 2017), we focus here on the loss-based updates for clarity. Since the encoder is kept frozen, both terms in practice serve to adapt the codebook vectors to the distribution of the bioacoustic embeddings, yielding a discrete vocabulary that best captures their statistical structure.

VQs are unfortunately also known to suffer from codebook collapse, where the codebook usage is highly imbalanced, i.e. most input embeddings get mapped to a one or two centroids, while the rest of the codebook remains idle and unpdated, drastically reducing its effective representation capacity.

9.3.2 Gumbel-Softmax Vector Quantization (GVQ)

To mitigate codebook collapse in the standard VQ, we also implement Gumbel Vector Quantization (GVQ) (Jang, S. Gu, and Poole, 2017), which uses the Gumbel–Softmax relaxation as a proxy for classic Softmax and to enable differentiable sampling from a categorical distribution. Given an input embedding $\boldsymbol{x}_n^{(l)}$, a linear projection layer computes logits $\{\pi_i\}_{i=1}^V$. The relaxed one-hot vector $\boldsymbol{p} \in \Delta^{V-1}$ is then obtained via:

$$p_i = \frac{\exp\big((\log \pi_i + g_i)/\tau\big)}{\sum_{j=1}^V \exp\big((\log \pi_j + g_j)/\tau\big)}, \tag{9.3}$$

where each g_i is an independent sample from the Gumbel$(0,1)$ distribution and τ is a fixed temperature (set to 1.0). A straight-through estimator is applied so that, during the forward pass, the highest-probability entry in $\boldsymbol{p}$ is discretized to a one-hot vector, while in the backward pass gradients flow through $\boldsymbol{p}$ as if the operation were identity.

Training of the GVQ module is driven by an entropy-maximizing loss that encourages uniform use of all V codewords. Equivalently, this can be written as a KL divergence between $\boldsymbol{p}$ and the uniform distribution:

$$\mathcal{L}_{\text{GVQ}} = \sum_{i=1}^{V} p_i \log\big(p_i\, V\big) \tag{9.4}$$

In our GVQ implementation, we implement several extensions to improve codebook utilization and robustness. First, we augment the KL divergence objective with a tunable weight

parameter α_{KL}. Second, we add a diversity loss term weighted by a hyperparameter λ_{div}, which explicitly penalizes under-utilization of the codebook. Throughout training, we track two key metrics: the codebook perplexity

$$\mathrm{PPL} = \exp\Big(-\sum_{i=1}^{V} \bar{p}_i \log \bar{p}_i\Big), \tag{9.5}$$

where $\bar{p}_i$ is the average probability of selecting codeword i, and the normalized perplexity PPL/V. The diversity loss is defined to increase the normalized perplexity, thereby encouraging the model to make use of a larger fraction of available codewords.

9.4 Experimental Setup

All of our experiments were conducted using the same preprocessing and batching pipeline to ensure a fair comparison across conditions. For this work, we stuck to HuBERT as our SSL model for extracting feature embeddings $\boldsymbol{x} \in \mathbb{R}^{L\times N\times D}$.

The remaining of this section is organized as follows: Section 9.4.1 gives an outline of the quantizer training protocol, and Section 9.4.2 provides the overview of the acoustic token generation, including the sequence post-processing.

9.4.1 Quantizer Training Protocol

We train all of our vector-quantization models on $\boldsymbol{x}$ using the Adam optimizer with a fixed batch size of 32, running for up to 20 epochs on *Train*, and evaluating performance on a held-out *Val* set to monitor convergence and guard against overfitting. To find the best hyperparameter settings, we conduct a grid search over two quantizer variants, as given in Table 9.1.

Table 9.1 – Hyperparameter search space for VQ and GVQ models.

Quantizer	Hyperparameter	Search Space
VQ	Learning rate	1e[-4, -3, -2]
	Commitment cost	0.25
	EMA	[True, False]
GVQ	Learning rate	1e[-4, -3, -2]
	KL weight	[0.5, 1.0, 1.5, 2.0]
	Diversity weight	[0.0, 0.01, 0.05, 0.1, 0.2, 0.5]
	Temperature schedule:	
	Max temperature	2.0
	Min temperature	0.1
	Decay factor	0.999

Note that for both quantizer models, the codebook $\mathcal{C}$ is *shared* across all layers L during training. Having the same symbol inventory for every layer makes the token sequences directly comparable across layers, and removes the need to have 13 separate vocabulary sets. Since the codebook must cover the union of all layer manifolds, a codebook-collapse is unlikely, and much less so than the alternate scenario of layer-specific sub-codebooks.

Each mini-batch therefore contains all layers of every utterance during training: batch tensors of shape (B, L, N, D), corresponding to the batch size, layer index, frame index, and feature dimension respectively, are reshaped to $(B \times L, N, D)$, quantized with a $V = 50$ entry codebook, and then reshaped back. This allows the quantizer q to see inputs from all layers, but then generate token sequences $\boldsymbol{t}$ drawn from the common symbol set.

9.4.2 Token Sequence Generation and Post-Processing

After training the quantizer on *Train*, we generate and save sequences of acoustic discrete tokens $\boldsymbol{t}$ for each vocalization in the entire dataset as described in the pipeline in Section 9.3. However, during batch processing, audio waveforms are repeat-padded to match the length of the longest sample within the batch. This repetition artificially inflates all the token sequences except one to be longer than the actual audio signals. To account for this, we apply some post-processing to the sequence by first calculating the effective number of frames of each data sample. We determine the downsampling factor of a batch by dividing the longest raw audio length in a given batch by the number of frames in its token sequence. Then, for each data sample, we compute the effective frame count by dividing its raw audio length by this factor and rounding the result. Finally, the token sequence for each sample is trimmed to this effective frame count, yielding a variable-length representation that accurately reflects the original signal duration and excludes any tokens that result solely from the padding. To ensure consistency with the original embedding extraction process, we implement verification mechanisms that confirm sample ordering is maintained throughout the token generation pipeline.

9.5 Distance Analysis

This section presents a distance analysis for the token sequences to identify any discernible patterns or correlations once we obtain the token sequences for each vocalization using the trained quantizers. Specifically, we are interested in observing the intra-class and inter-class variability to understand the degree with which the generated token sequences are able to distinguish from one class to another.

We use the Levenshtein distance $d(\boldsymbol{t}_1, \boldsymbol{t}_2)$, a string metric also known as the edit distance, to quantitatively measure the distance between a pair of discrete token sequences $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$. The distance effectively represents the minimum number of 'edits', i.e. insertions, deletions, or substitutions, needed to change one sequence into the other. A distance $d = 0$ thus means

that the two sequences are identical. It can go up to at most the length of the longer string. However, this metric gives an absolute difference between sequences and is misrepresentative when a pair of sequences have a large difference in lengths. To overcome this issue, we use the normalized Levenshtein distance, which divides the calculated distance by the length of the longer sequence $\frac{d(\boldsymbol{t}_1,\boldsymbol{t}_2)}{\max(|\boldsymbol{t}_1|,|\boldsymbol{t}_2|)}$, where $|\cdot|$ denotes the length of the sequence. In this case, the distance is bounded between 0 and 1, representing identical and completely different sequences respectively. In the case of $d = 1$, one need to edit every character in the longer string to transform it into the other.

We compute the pairwise Levenshtein distances for all data samples, grouping each comparison into one of the following four possible permutations:

(i •) *Intra-caller, intra-calltype*: two vocalization samples from the same caller producing the same call-type. The distance between these is expected be the smallest.

(ii •) *Intra-caller, inter-calltype*: two vocalization samples from the same caller producing different call-type.

(iii •) *Inter-caller, intra-calltype*: two vocalizations from different callers producing the same call-type.

(iv •) *Inter-caller, inter-calltype*: two vocalizations from different callers producing different call-types. The distance between these is expected be the largest.

Figure 9.2 presents the means of the distances distributions of the four aforementioned categories, using the token sequences generated from the VQ model. We can observe that groups (i •) and (iv •) behave as expected: they both have the smallest and largest distance, on average, for all datasets. We also noticeably observe that group (ii •)'s distance is larger than group (iii •)'s for most datasets. This makes sense intuitively: two vocalizations produced by the a caller vocalizing different call-types are more likely to be acoustically distinct, than two generated by different callers vocalizing the same call-type. The discrete acoustic tokens sequences reflect this distribution, demonstrating their ability to model and capture the temporal information encoded in vocalizations.

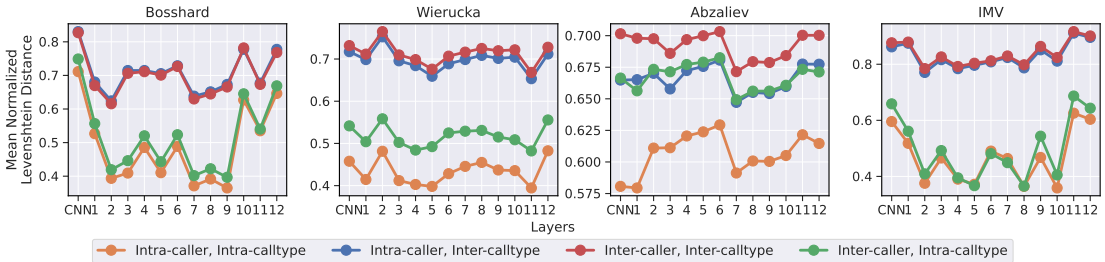


Figure 9.2 – Layer-wise mean Levenshtein distance between all pairs of VQ token sequences.

While we can observe similar trends with the GVQ tokens on the Bosshard dataset, as shown on Figure 9.3, the remaining datasets exhibit some different patterns. Notably, group (ii •) and (iii •)'s trends are flipped in the Abzaliev dataset, showing that the intra-caller, inter-calltype distances are smaller than inter-caller, intra-calltype ones. This may be due to the comparatively large number of callers (80), which increases acoustic variability and makes it harder to distinguish sequences of the same call-type produced by different callers than those of different call-types produced by the same caller. Additionally, for Wierucka and IMV datasets, the pairwise distances in group (iii •) are unexpectedly smaller on average than in group (i •). This suggests that the GVQ tokens do not consistently preserve fine-grained caller-specific information as well as the VQ tokens across all datasets.

Taken together, this analysis indicates that the standard VQ discrete token representations are indeed capable of clustering sufficient acoustic information to discriminate by call-type or by caller identity, under real-world, left-to-right temporal constraints. The degree of separability can be measured with a token sequence classification task. The GVQ tokens, however, exhibit some unexpected patterns and less consistent separability, indicating that they may be less effective for classification.

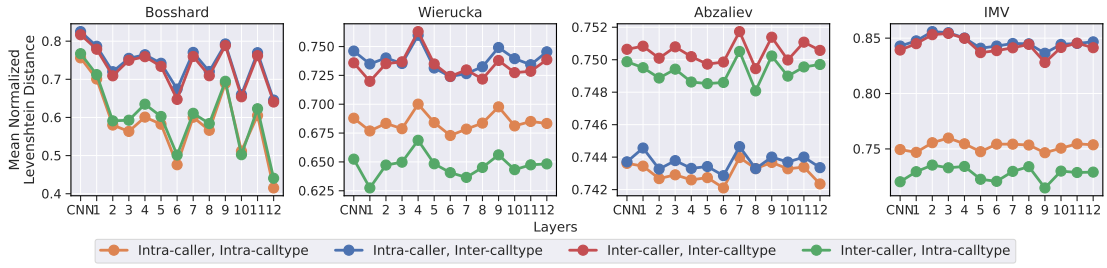


Figure 9.3 – Layer-wise mean Levenshtein distance between all pairs of GVQ token sequences.

9.6 Classification Analysis

Based on the insights of the comparative analysis, in section, we evaluate how well the sequential nature of token representations can be leveraged for call-type (CTID) and caller (CLID) classification.

9.6.1 Experimental Setup

We classify the token sequences using the k-Nearest Neighbours (k-NN) algorithm. We use the pre-computed pairwise Levenshtein distances as our distance similarity matrix, and iterate over the hyperparameters given in Table 9.2, for each layer, to obtain optimal classification results. The classifier is trained over *Train*, and the hyperparameters defined in the search space are evaluted over *Val*, using UAR as the optimization criterion. The best hyperparameters are then used on *Test*. The predicted label of a sample is determined by applying a majority-voting framework on the actual labels of the k most similar sequences.

Table 9.2 – Hyperparameter search space used for training the k-NN classifier.

Classifier	Hyperparameter	Search Space
k-NN	Number of neighbours k	[1, 3, 5, 7, 9]
	Neighbour weighting	[Uniform, distance]
	Distance	Levenshtein
	Task	[CTID, CLID]

9.6.2 Results and Discussion

The CTID results are shown in Figure 9.4 for the VQ and GVQ token sequences. We compare the results to a neural linear-probing baseline, as employed in the previous chapters, i.e. by pooling the temporal information into a functional vector $\boldsymbol{f}_{\sigma\mu} \in \mathbb{R}^{2D}$, and classifying it using a fully-connected layer.

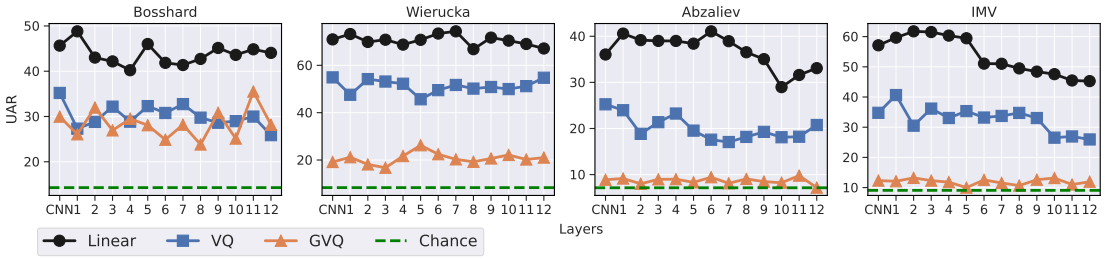


Figure 9.4 – Layer-wise UAR [%] for CTID using k-NN on token sequences

We can observe that the linear layer clearly outperforms the k-NN-based classification of token sequences across all four datasets. Surprisingly, the VQ representations also consistently and substantially perform better than the GVQ ones for all datasets except Bosshard. The same trend can again observed for the CLID task, shown on Figure 9.5. GVQ especially struggles on the Abzaliev dataset, essentially achieving chance-level performance. This strongly suggests that the GVQ codebook has converged to a local optimum, or potentially collapsed to only a small subset of symbols. In addition, while a single shared codebook can still encode enough information for call–type discrimination, it is perhaps not expressive enough to preserve the finer caller-specific nuances that exist in the continuous embeddings.

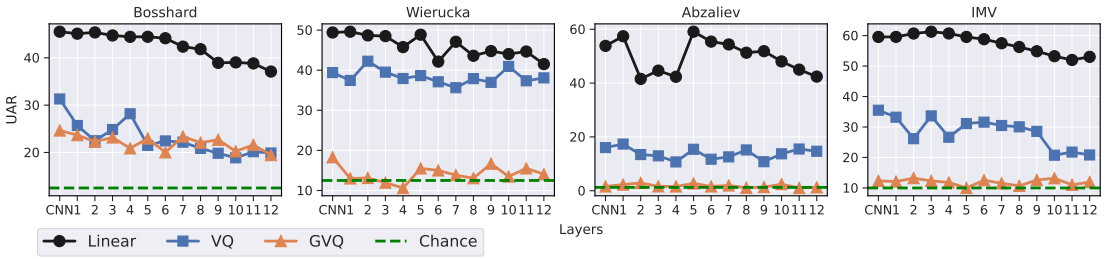


Figure 9.5 – Layer-wise UAR [%] for CLID using k-NN on token sequences.

The overall trends clearly indicate that while discrete token sequences do carry phonotactic information that can be leveraged, the HuBERT-based feature embeddings still capture much more meaningful information, even when stats-pooled into a vocalization-level vector. In other words, the data tokenization process of converting the feature embeddings causes a higher loss of information than what is gained by keeping and leveraging the temporal structure of vocalizations at token-level representations.

Although we trained a single codebook, shared across all layers, for both VQ and GVQ, we still observe that earlier layers tend to yield better performance across tasks, consistent with the trends reported in previous chapters. This indicates that differences between layers persist even after discretization, and that sharing a codebook does not diminish the higher capability of earlier layers in encoding salient and transferable representations.

Table 9.3 – Best UAR [%] scores for each feature across layers. n_C is the number of classes for that dataset and task, and chance performance is calculated as $100/n_c$. **Δ** represents the relative drop in performance with respect to the linear layer baseline.

Task	Dataset	n_C	Chance	Linear	VQ	GVQ	ΔVQ	ΔGVQ
CTID	Bosshard	7	14.30	48.81	35.20	35.52	27.88	27.23
	Wierucka	12	8.30	74.36	54.91	26.23	26.16	64.72
	Abzaliev	14	7.14	41.07	25.24	9.78	38.54	76.20
	IMV	11	9.10	61.75	40.65	24.94	34.17	59.60
CLID	Bosshard	8	12.50	45.52	31.31	24.65	31.22	45.85
	Wierucka	8	12.50	49.60	42.24	18.29	14.83	63.13
	Abzaliev	80	1.25	59.09	17.35	2.90	70.64	95.09
	IMV	10	10.00	61.28	35.51	13.23	42.05	78.42

Table 9.3 tabulates the highest scores of each feature across layers, and also shows the drop in performance, denoted with Δ, of the token sequence-based representations compared to the linear baseline. Similar to the results in previous chapters, we can see that the CTID classification yields higher scores than CLID across all feature representations. This highlights that call-types differ in distinct spectro-temporal patterns that token sequences can still capture, where as caller identity is largely carried by subtler characteristics that are harder to preserve after vector quantization. This also suggests that discrete tokens need a higher-resolution to be effective.

Figure 9.6 visually plots the same information. For CTID, discretizing the feature embeddings with a VQ and GVQ drops the performance across datasets by ~26-39% and ~27-79% respectively, when compared to stats-pooling the same features and then classifying with a linear layer. For CLID, the drop is of ~15-71% and ~46-95% respectively. These strong decreases in performances reveal that perhaps a single VQ or GVQ codebook is not enough to effectively model the entire animal vocalizations alone, especially for CLID, or the arbitrary codebook size of $V = 50$. In our early ablation experiments, however, we did not empirically observe a significant change in performance when compared to $V = 25$ or 100. A plausible next step

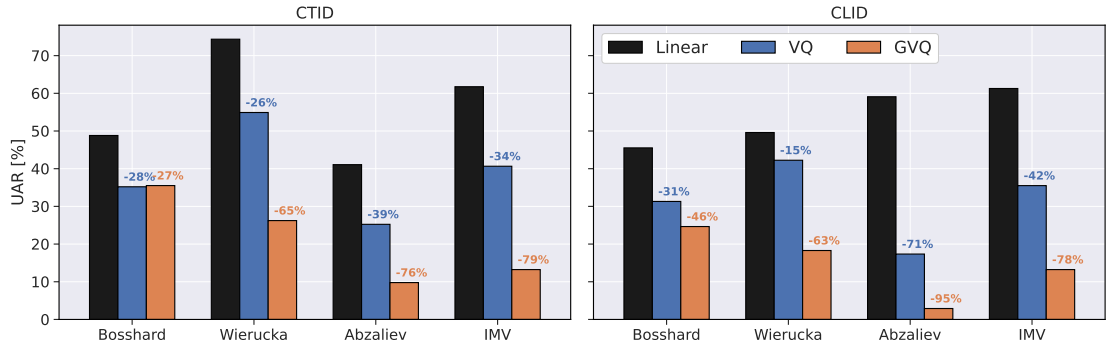


Figure 9.6 – Best UAR results across layers for CTID and CLID.

could thus be to train a quantizer model which employs *multiple* codebooks to retain a richer set of temporal patterns.

9.7 Conclusions and Future Work

In this chapter, we explored alternate feature representations that could preserve the temporal structure of animal vocalizations instead of averaging their extracted SSL feature embeddings into single functional vectors, as in previous chapters. To that end, we investigated and evaluated whether discrete acoustic token-based feature representations could effectively improve call-type and caller classification performance.

To address this problem, we first trained a conventional vector quantization and a Gumbel-softmax vector quantization module to convert the vocalization signals into discrete token sequences for four different animal datasets. In our initial line of investigation, we conducted a comparative analysis of the generated sequences using the Levenshtein distance metric. The results showed that they do encode the sequential structure of animal calls, and exhibit a degree of separability by call-type or caller identity across all datasets. We then trained a k-Nearest Neighbour classifier on said representations to evaluate how well they could systematically distinguish vocalizations by call-type and caller identity. The results showed that both representations were significantly weaker than a simple linear-probe baseline for all datasets. While VQ showed a reasonable performance, GVQ yielded poor scores, nearing chance level in many cases. Overall, the results indicate that token sequences do encode meaningful sequential structure, but the information lost during vector quantization outweighs the benefits gained from explicit temporal modeling.

The scope for improvements on this topic is fairly large. A direct line of investigation would be to improve the quantization module to reduce the information loss. Future work should explore larger, multi-codebook quantization architectures, such as Residual VQs (Juang and Gray, 1982) or Grouped VQ (Jégou, Douze, and Schmid, 2011). Wav2Vec2 notably employs a grouped VQ module with $G = 2$ codebooks of size $V = 320$. Knowing that its feature embeddings gave a similar performance to HuBERT for Marmoset caller detection in Chapter 4 (Sarkar and

Magimai.-Doss, 2023), evaluating its token sequences against a matched linear-probe baseline could give meaningful insights.

Another direction of future work could explore more sequence post-processing techniques, such as deduplication, i.e. removing consecutive duplicate tokens (X. Chang et al., 2024), or acoustic byte-pair encoding (BPE) (Gage, 1994). These can further reduce the sequence length and tighten the alignment between tokens and acoustically meaningful sub-units, which could be particularly useful for vocalizations whose acoustic structure changes slowly.

In summary, despite the promise of symbol-based sequence modeling, this chapter confirms that simple stats-pooled functional vectors remain a highly effective representation for bioacoustic classification tasks, even though they don't directly leverage the temporal structure of animal vocalizations.

10 Conclusions and Future Directions

10.1 Conclusions

Non-human animal vocalizations encode a wide range of information, such as call-type and caller identity. This thesis studied the transfer of representations learnt from human speech, in a self-supervised learning framework, to decode animal vocal communication. We focused our investigations on a handful of research questions addressed across the different chapters of this thesis, using vocalizations of dogs, marine mammals, and especially marmosets.

We first investigated the notion of speech-to-vocalizations cross-transferability and discussed the potential of domain-agnostic pre-training of speech self-supervised learning (SSL) models for decoding animal calls. We argued that, since these models use only the intrinsic structure of a given input signal to extract essential information onto an embedding space, independent of its acoustic domain, their utility should not be limited to modeling human speech alone. Building on this understanding, we conducted a caller detection study on marmoset vocalizations as a proof-of-concept, using eleven pre-trained SSL models and the InfantMarmosetsVox dataset. Our results showed that the embedding spaces did carry meaningful caller information, and enabled us to systematically and successfully distinguish individual identities of marmoset in a binary classification framework without any downstream fine-tuning. This was the first study to demonstrate that human speech-learnt representations transfer to non-human animal vocalizations – a finding that has since been further corroborated on other taxa, such as gibbons (Cauzinille et al., 2024) or bats (Heer Kloots and Knornschild, 2024).

We also extended and validated our approach beyond a binary caller detection task on a single dataset, to multi-class call-type classification, caller identification, and caller sex identification across multiple datasets. In addition to SSLs, we explored alternate feature representations, namely an end-to-end acoustic model, and a hand-crafted Catch22 baseline. Through comprehensive experiments, we demonstrated that SSL-based feature representations and end-to-end acoustic modeling led to better systems than Catch22 features for call-type and caller classification, and achieved comparable performances for sex identification at an identical sampling rate. Furthermore, we observed that the lower SSL layers were much more salient

representations and yielded higher scores for all three tasks across all datasets than the higher layers.

We also shifted our perspective from evaluating performance across tasks and datasets, to scrutinizing and questioning the utility in applying 'off-the-shelf' SSL models to marmoset call analysis, and by extension animal vocalizations in general. SSL models are typically pre-trained at a bandwidth of 8 kHz, which mismatches with the higher-frequency acoustic vocalizations and auditory range of marmosets, leading to a significant loss of biologically relevant information. Our experiments revealed that increasing the bandwidth size yields a monotonic improvement in classification performance, highlighting that pre-trained SSL models can be highly effective for bioacoustic tasks, provided their training bandwidth aligns with the vocal frequency range of the target species.

As the field evolved while this thesis was in progress, a new generation of models pre-trained directly on bioacoustic data began to appear, outperforming strong baselines across animal benchmarks (Hagiwara, 2023a). We explored whether these specialized models actually offered a significant advantage over those pre-trained on speech. Surprisingly, the head-to-head comparison results showed that bioacoustics-trained models only yielded marginal gains in a few select contexts, and otherwise matched the performance of speech-pretrained networks. In addition, it was also unclear how models pre-trained on human speech compared to those trained on general audio. Results showed that general audio performed comparably to those pre-trained on speech, suggesting that it is the domain-agnostic self-supervised pre-training itself, i.e. the way the model is encouraged to discover intrinsic structure in any audio signal, rather than the specific acoustic domain, that endows these networks with cross-domain generalizability.

Beyond training classifiers on features extracted from frozen pre-trained models, we also investigated directly fine-tuning them. First, we investigated whether fine-tuning speech pre-trained models on automatic speech recognition (ASR) task in a supervised framework could introduce an inductive bias, enhancing them for bioacoustic classification. However, this produced mixed results, offering no consistent improvement, suggesting that the general-purpose representations learned during SSL pre-training were already well-suited for bioacoustic tasks. We then explored whether fine-tuning pre-trained speech or bioacoustics SSL models directly on the downstream animal data would yield better performance. We demonstrated that these models can be successfully adapted to improve call-type classification performance when ample labeled data is available, and can substantially improve performance compared to a simple linear classifier.

Finally, we looked at alternate feature representations which could preserve the sequential structure of animal vocalizations, instead of pooling them into a single functional vector, and leverage the encoded temporal information to improve performance on call-type and caller identity classification. We trained vector quantizers transform extracted feature embeddings into discrete acoustic token sequences, and then classify them using a k-Nearest

Neighbour classifier. However, the results showed that token-based feature representations were substantially weaker for both tasks, than a simple linear-layer applied to the stats-pooled functional vector. This highlighted the latter's effectiveness as a feature representation for animal vocalization classification tasks.

Taken together, these studies establish that self-supervised speech, as well general audio classification models, constitute a powerful, domain-agnostic toolkit and offer a remarkably versatile starting point for decoding non-human animal vocal communication. This thesis provides a practical and robust framework for advancing bioacoustic analysis, that can be readily extended to new species, recording conditions, and behavioral contexts. As SSL models continue to evolve, our framework and findings point the way toward increasingly sensitive, scalable bioacoustic systems with minimal species-specific feature engineering.

10.2 Limitations and Future Directions

Most contributions of this thesis have been at a foundational-level, focusing on leveraging technologies developed for human speech, and demonstrating their feasibility or adaptability for non-human animal vocalizations. The next step would be to leverage these findings to develop these frameworks from proof-of-concepts into full applications and tools for further research. There are several directions that could be expanded on for future research:

Automated vocalization detection: in this thesis, we always assumed pre-segmented calls, and did not investigate the detection of animal vocalizations within audio recordings. Developing robust call detection methods which can work in-the-wild are of particular interest, as bioacoustics data comes from challenging, un-controlled, and noisy environments, and domain experts in animal calls are very rare. Developing robust automated vocalization detection systems is of significant value as it can vastly reduce the amount of manual expert interventions needed. The same SSLs feature embeddings can likely be very easily leveraged for this task. Alternate unsupervised and computationally efficient signal-processing based techniques could also advantageous over deep learning-based methods as they are often more interpretable for linguistic research.

Caller diarization in multi-speaker recordings: progressing from animal caller identity classification to full caller diarization would be extremely valuable to researchers collecting data with multiple individuals vocalizing in the same audio recording. Human speech diarization is large field with a rich history. Off-the-shelf speech-diarization frameworks could likely provide a starting point, but adapting to animal-specific needs will provide significant value and benefit.

Linking tokens to acoustic and biological correlates: If more complex quantizers, such as residual or grouped VQs, can surpass linear-probing baselines, one could leverage discrete acoustic tokens to develop an inventory of sub-vocalization units per species. Such a token lexicon could open up novel research directions in computational bioacoustics, particularly

in understanding combinatorial structure within animal calls, and drawing closer parallels with phonemic organization in human language. However, a more detailed acoustic and spectral investigation is needed to meaningfully ground these discrete units in biology. This includes analyzing their spectral and temporal properties, such as pitch, duration, vocal tract resonances, and harmonic structure, and identifying how these potentially relate to known physiological or behavioral correlates. Doing so could help clarify whether particular tokens correspond to biologically meaningful units, such as arousal, social intent, or vocal production mechanisms. Future work could also consider species-specific signal analyses, as articulatory constraints and communicative functions vary widely across species.

Cross-species transfer and generalisation: in this thesis we focused on the transferability of human speech or general audio to animal vocalizations. Cross-species transfer remains an open-question. Evaluating how well SSL models or quantizers trained on one species transfer to others could give reveal the broader applicability of feature embeddings or symbolic audio representations.

Bibliography

Abzaliev, A., H. Perez-Espinosa, and R. Mihalcea (May 2024). "Towards Dog Bark Decoding: Leveraging Human Speech Processing for Automated Bark Classification". In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Ed. by N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue. Torino, Italia: ELRA and ICCL (cit. on pp. 25, 56, 58).

Agamaite, J. A., C. J. Chang, M. S. Osmanski, and X. Wang (2015). "A quantitative acoustic analysis of the vocal repertoire of the common marmoset (Callithrix jacchus)". In: *The Journal of the Acoustical Society of America* 138(5), pp. 2906–2928 (cit. on pp. 22, 24, 40, 42, 43, 45).

Agamaite, J., C. Chang, M. Osmanski, and X. Wang (Nov. 2015). "A quantitative acoustic analysis of the vocal repertoire of the common marmoset (Callithrix jacchus)". English (US). In: *Journal of the Acoustical Society of America* 138.5, pp. 2906–2928 (cit. on p. 53).

Aghajanyan, A., S. Gupta, and L. Zettlemoyer (Aug. 2021). "Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by C. Zong, F. Xia, W. Li, and R. Navigli. Online: Association for Computational Linguistics, pp. 7319–7328 (cit. on p. 66).

Atal, B. S. and S. L. Hanauer (1971). "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave". In: *Proceedings of the IEEE*. Vol. 63. 4, pp. 561–580 (cit. on p. 1).

Baevski, A., W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli (July 2022). "data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language". In: *Proc. of ICML*. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 1298–1312 (cit. on pp. 28, 31).

Baevski, A., S. Schneider, and M. Auli (2020). "vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations". In: *International Conference on Learning Representations* (cit. on p. 83).

Baevski, A., Y. Zhou, A. Mohamed, and M. Auli (2020). "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in Neural Information Processing Systems* 33, pp. 12449–12460 (cit. on pp. 1, 15, 28, 31, 59, 83).

Bai, Z. and X.-L. Zhang (2021). "Speaker recognition based on deep learning: An overview". In: *Neural Networks* 140, pp. 65–99 (cit. on p. 1).

Banville, H., I. Albuquerque, A. Hyvärinen, G. Moffat, D.-A. Engemann, and A. Gramfort (2019). "Self-Supervised Representation Learning from Electroencephalography Signals". In: *International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6 (cit. on p. 28).

Banville, H., O. Chehab, A. Hyvärinen, D.-A. Engemann, and A. Gramfort (Mar. 2021). "Uncovering the structure of clinical EEG signals with self-supervised learning". In: *Journal of Neural Engineering* 18.4, p. 046020 (cit. on p. 28).

Bengio, Y., N. Léonard, and A. Courville (2013). *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation* (cit. on p. 84).

Bergler, C. et al. (2019). "ORCA-SPOT: An automatic killer whale sound detection toolkit using deep learning". In: *Scientific Reports* 9.1, pp. 1–10 (cit. on p. 2).

Bermant, P. C., L. Brickson, and A. J. Titus (2022a). "Bioacoustic Event Detection with Self-Supervised Contrastive Learning". In: *bioRxiv* (cit. on p. 23).

Bermant, P. C., L. Brickson, and A. J. Titus (2022b). "Bioacoustic Event Detection with Self-Supervised Contrastive Learning". In: *bioRxiv* (cit. on p. 28).

Bezerra, B. and A. Souto (June 2008). "Structure and Usage of the Vocal Repertoire of Callithrix jacchus". In: *International Journal of Primatology* 29, pp. 671–701 (cit. on p. 22).

Bhattacharyya, A. (1943). "On a Measure of Divergence between Two Statistical Populations Defined by Their Probability Distributions". In: *Bulletin of the Calcutta Mathematical Society* 33, pp. 99–109 (cit. on p. 31).

Blumstein, D. T. et al. (2011). *Acoustic monitoring in terrestrial environments using microphone arrays: Applications, technological considerations, and prospectus*. Journal of Applied Ecology (cit. on p. 1).

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, pp. 144–152 (cit. on p. 17).

Bosshard, A. B. (2020). "Sequential dynamics in common marmoset vocal strings". Master's Thesis. University of Zurich (cit. on p. 24).

Bosshard, A. B., J. M. Burkart, P. Merlo, C. Cathcart, S. W. Townsend, and B. Bickel (2024). "Beyond bigrams: call sequencing in the common marmoset (Callithrix jacchus) vocal system". In: *Royal Society Open Science* 11.11, p. 240218 (cit. on p. 24).

Bradbury, J. W. and S. L. Vehrencamp (1998). *Principles of Animal Communication*. Sinauer Associates, Sunderland, MA (cit. on p. 1).

Breiman, L. (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32 (cit. on p. 17).

Brumm, H., K. Voss, I. Köllmer, and D. Todt (Jan. 2004). "Acoustic communication in noise: regulation of call characteristics in a New World monkey". In: *Journal of Experimental Biology* 207.3, pp. 443–448 (cit. on p. 22).

BS, J., H. DHR, and C. CK (1993). "The stability of the vocal signature in phee calls of the common marmoset, Callithrix jacchus". In: *American journal of primatology* 31(1), pp. 67–75 (cit. on p. 22).

Burkart, J. M., J. E. C. Adriaense, R. K. Brügger, F. M. Miss, K. Wierucka, and C. P. van Schaik (2022). "A convergent interaction engine: vocal communication among marmoset monkeys". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 377.1859, p. 20210098 (cit. on p. 22).

Catchpole, C. K. and P. J. B. Slater (2003). *Bird Song: Biological Themes and Variations*. Cambridge: Cambridge University Press (cit. on p. 82).

Cauzinille, J., B. Favre, R. Marxer, D. Clink, A. H. Ahmad, and A. Rey (2024). "Investigating self-supervised speech models' ability to classify animal vocalizations: The case of gibbon's vocal signatures". In: *Proc. of Interspeech* (cit. on pp. 56, 95).

Chang, H.-J., S.-w. Yang, and H.-y. Lee (2022). "DistilHuBERT: Speech representation learning by layer-wise distillation of hidden-unit BERT". In: *Proc. of ICASSP*. IEEE, pp. 7087–7091 (cit. on pp. 28, 31).

Chang, X. et al. (2024). "Exploring Speech Recognition, Translation, and Understanding with Discrete Speech Units: A Comparative Study". In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11481–11485 (cit. on p. 93).

Chen, H., W. Xie, A. Vedaldi, and A. Zisserman (2020). "Vggsound: A Large-Scale Audio-Visual Dataset". In: *Proc. of ICASSP*, pp. 721–725 (cit. on pp. 15, 59).

Chen, S. et al. (2022). "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing". In: *IEEE Journal of Selected Topics in Signal Processing* 16.6, pp. 1505–1518 (cit. on pp. xii, 15, 28, 31, 41, 42, 44, 50, 59).

Chung, Y.-A., W.-N. Hsu, H. Tang, and J. Glass (2019). "An unsupervised autoregressive model for speech representation learning". In: *Proc. of Interspeech* (cit. on pp. 14, 28, 31).

Chung, Y.-A., H. Tang, and J. Glass (2020). "Vector-quantized autoregressive predictive coding". In: *Proc. of Interspeech* (cit. on pp. 14, 28, 31).

Coffey, E. et al. (2019). "Deep representation learning for orca call type classification". In: *Scientific Reports* 9.1, pp. 1–10 (cit. on p. 2).

Coffey, K. R., R. G. Marx, and J. F. Neumaier (2019). "DeepSqueak: a deep learning-based system for detection and analysis of ultrasonic vocalizations". In: *Neuropsychopharmacology* 44, pp. 859–868 (cit. on p. 46).

Cortes, C. and V. Vapnik (Sept. 1995). "Support-vector networks". In: *Machine Learning* 20.3, pp. 273–297 (cit. on p. 16).

Dahl, G. E., D. Yu, L. Deng, and A. Acero (2012). "Context-Dependent Pre-trained Deep Neural Networks for Large-Vocabulary Speech Recognition". In: *IEEE Transactions on Audio, Speech, and Language Processing*. Vol. 20. 1, pp. 30–42 (cit. on p. 1).

Davis, S. B. and P. Mermelstein (1980). "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing*. Vol. 28. 4, pp. 357–366 (cit. on p. 1).

Dehak, N., P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet (2011). "Front-End Factor Analysis for Speaker Verification". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4, pp. 788–798 (cit. on p. 30).

Den Oord, A. van, O. Vinyals, and K. Kavukcuoglu (2017). "Neural discrete representation learning". In: *Advances in Neural Information Processing Systems* (cit. on pp. 84, 85).

Denton, T. (2023). *Google Bird Vocalization Classifier: A global bird embedding and classification model (Perch)*. https://github.com/google-research/perch. (Accessed on 09/12/2024) (cit. on p. 56).

Desplanques, B., J. Thienpondt, and K. Demuynck (2020). "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification". In: *Interspeech 2020*, pp. 3830–3834 (cit. on p. 1).

Dubagunta, S. P., B. Vlasenko, and M. Magimai.-Doss (2019). "Learning voice source related information for depression detection". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (cit. on p. 41).

Dufourq, E., C. Batist, R. Foquet, and I. Durbach (2022). "Passive acoustic monitoring of animal populations with transfer learning". In: *Ecological Informatics* 70, p. 101688 (cit. on p. 56).

Durrieu, J.-L., J.-P. Thiran, and F. Kelly (2012). "Lower and upper bounds for approximation of the Kullback-Leibler divergence between Gaussian Mixture Models". In: *Proc. of ICASSP* (cit. on p. 31).

Eliades, S. J. and C. T. Miller (2017). "Marmoset vocal communication: Behavior and neurobiology". In: *Developmental Neurobiology* 77.3, pp. 286–299 (cit. on p. 22).

Eliades, S. J. and X. Wang (2012). "Neural Correlates of the Lombard Effect in Primate Auditory Cortex". In: *Journal of Neuroscience* 32.31, pp. 10737–10748 (cit. on p. 22).

Epple, G. (1968). "Comparative studies on vocalization in marmoset monkeys (Hapalidae)". In: *Folia Primatol (Basel)* 8.1, pp. 1–40 (cit. on p. 22).

Fant, G. (1970). *Acoustic Theory of Speech Production*. The Hague: Mouton (cit. on p. 1).

Fedurek, P., K. E. Slocombe, and K. Zuberbühler (2016). "Sequential information in a great ape utterance". In: *Scientific Reports* 6, p. 38226 (cit. on p. 1).

Fitch, W. T. (2018). "The biology and evolution of speech: A comparative analysis". In: *Annual Review of Linguistics* 4, pp. 255–279 (cit. on p. 1).

Fonseca, E., X. Favory, J. Pons, F. Font, and X. Serra (2021). "Fsd50k: an open dataset of human-labeled sound events". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (cit. on pp. 15, 59).

Freund, Y. and R. E. Schapire (1997). "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *Journal of Computer and System Sciences* 55.1, pp. 119–139 (cit. on p. 17).

Fulcher, B. D., M. A. Little, and N. S. Jones (2013). "Highly comparative time-series analysis: the empirical structure of time series and their methods". In: *Journal of The Royal Society Interface* 10.83 (cit. on pp. 11, 39, 49).

Gage, P. (Feb. 1994). "A new algorithm for data compression". In: *C Users J.* 12.2, pp. 23–38 (cit. on p. 93).

Gemmeke, J. F. et al. (2017). "Audio set: An ontology and human-labeled dataset for audio events". In: *Proc. of ICASSP* (cit. on pp. 15, 59).

Ghani, B., T. Denton, S. Kahl, and H. Klinck (2023). "Global birdsong embeddings enable superior transfer learning for bioacoustic classification". In: *Scientific Reports* 13.1, p. 22876 (cit. on pp. 2, 56).

Ghazanfar, A. A. and D. Rendall (2008). "Evolution of human vocal production". In: *Current Biology* 18.11, R457–R460 (cit. on p. 2).

Goffinet, J., S. Brudner, R. Mooney, and J. Pearson (May 2021). "Low-dimensional learned feature spaces quantify individual and group differences in vocal repertoires". In: *eLife* 10. Ed. by J. H. Goldberg, T. E. Behrens, J. H. Goldberg, O. Tchernichovski, and S. W. Linderman, e67855 (cit. on p. 46).

Graves, A., A.-r. Mohamed, and G. Hinton (2013). "Speech Recognition with Deep Recurrent Neural Networks". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649 (cit. on p. 1).

Gu, N., K. Lee, M. Basha, S. Kumar Ram, G. You, and R. H. R. Hahnloser (2024). "Positive Transfer of the Whisper Speech Transformer to Human and Animal Voice Activity Detection". In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7505–7509 (cit. on p. 46).

Guo, Y. et al. (2025). *Recent Advances in Discrete Speech Tokens: A Review* (cit. on p. 82).

Hagiwara, M. (2023a). "AVES: Animal Vocalization Encoder Based on Self-Supervision". In: *Proc. of ICASSP*, pp. 1–5 (cit. on pp. 15, 25, 56, 57, 59, 96).

Hagiwara, M., B. Hoffman, J.-Y. Liu, M. Cusimano, F. Effenberger, and K. Zacarian (2023b). "BEANS: The Benchmark of Animal Sounds". In: *Proc. of ICASSP*, pp. 1–5 (cit. on pp. 25, 56, 57).

Hauser, M. D. (1996). *The Evolution of Communication*. MIT Press (cit. on p. 1).

Heer Kloots, M. de and M. Knornschild (2024). "Exploring bat song syllable representations in self-supervised audio encoders". In: *Proc. of 4th International Workshop on Vocal Interactivity in-and-between Humans, Animals and Robots (VIHAR)* (cit. on pp. 56, 95).

Heggan, C., S. Budgett, T. Hospedales, and M. Yaghoobi (2024). "On the Transferability of Large-Scale Self-Supervision to Few-Shot Audio Classification". In: *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 515–519 (cit. on p. 56).

Hinton, G. et al. (2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition". In: *IEEE Signal Processing Magazine* 29.6, pp. 82–97 (cit. on p. 1).

Hornik, K., M. Stinchcombe, and H. White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5, pp. 359–366 (cit. on p. 8).

Hsu, C.-W. and C.-J. Lin (2002). "A comparison of methods for multiclass support vector machines". In: *IEEE Transactions on Neural Networks* 13.2, pp. 415–425 (cit. on p. 17).

Hsu, W.-N., B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed (2021). "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, pp. 3451–3460 (cit. on pp. 1, 15, 28, 31, 59, 83).

Hu, E. J. et al. (2022). "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations* (cit. on p. 66).

Huang, J., H. Ma, Y. Sun, L. Chang, and N. Gong (2022). “Complex rules of vocal sequencing in marmoset monkeys”. In: *bioRxiv* (cit. on p. 30).

Hurford, J. R. (2012). *The Origins of Grammar: Language in the Light of Evolution II*. Oxford: Oxford University Press (cit. on p. 1).

Hutter, F., H. Hoos, and K. Leyton-Brown (June 2014). “An Efficient Approach for Assessing Hyperparameter Importance”. In: *Proceedings of International Conference on Machine Learning 2014 (ICML 2014)*, pp. 754–762 (cit. on p. 74).

J, B. and L. JAM (2018). “Ultrasonic components of vocalizations in marmosets”. In: *Handbook of ultrasonic vocalization (ed. S Brudzynski)*, pp. 535–544 (cit. on p. 22).

Jang, E., S. Gu, and B. Poole (2017). “Categorical Reparameterization with Gumbel-Softmax”. In: *International Conference on Learning Representations* (cit. on p. 85).

Jégou, H., M. Douze, and C. Schmid (2011). “Product Quantization for Nearest Neighbor Search”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.1, pp. 117–128 (cit. on p. 92).

Juang, B.-H. and A. Gray (1982). “Multiple stage vector quantization for speech coding”. In: *ICASSP ’82. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 7, pp. 597–600 (cit. on p. 92).

Jurafsky, D. and J. H. Martin (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. Online manuscript released January 12, 2025 (cit. on p. 2).

Kahl, S., C. M. Wood, M. Eibl, and H. Klinck (2021). “BirdNET: A deep learning solution for avian diversity monitoring”. In: *Ecological Informatics* 61, p. 101236 (cit. on pp. 2, 56).

Kershenbaum, A., D. Blumstein, et al. (2016). “Acoustic sequences in non-human animals: a tutorial review and prospectus”. In: *Biological Reviews* 91.1, pp. 13–52 (cit. on p. 81).

Kershenbaum, A., A. Ilany, L. Blaustein, and E. Geffen (2012). “Syntactic structure and geographical dialects in the songs of male rock hyraxes”. In: *Proceedings of the Royal Society B: Biological Sciences* 279, pp. 2974–2981 (cit. on p. 82).

Kershenbaum, A., D. T. Blumstein, et al. (2016). “Acoustic sequences in non-human animals: a tutorial review and prospectus”. In: *Biological Reviews* 91.1, pp. 13–52 (cit. on p. 82).

Knight, E. et al. (2024). “Individual identification in acoustic recordings”. In: *Trends in Ecology & Evolution* (cit. on p. 56).

Kong, Q., Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley (2020). “PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28, pp. 2880–2894 (cit. on pp. 12, 44, 50).

LeCun, Y. et al. (1989). “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4, pp. 541–551 (cit. on p. 9).

LeCun, Y. and Y. Bengio (1998). “Convolutional networks for images, speech, and time series”. In: *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, pp. 255–258 (cit. on p. 9).

LeCun, Y., Y. Bengio, and G. Hinton (May 2015). “Deep learning”. In: *Nature* 521.7553, pp. 436–444 (cit. on p. 1).

Liu, A. H., Y.-A. Chung, and J. Glass (2021). “Non-Autoregressive Predictive Coding for Learning Speech Representations from Local Dependencies”. In: *Proc. of Interspeech*, pp. 3730–3734 (cit. on pp. 15, 28, 31).

Liu, A. T., S.-W. Li, and H.-y. Lee (July 2021). “TERA: Self-Supervised Learning of Transformer Encoder Representation for Speech”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, pp. 2351–2366 (cit. on pp. 15, 28, 31).

Liu, A. T., S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee (2020). “Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders”. In: *Proc. of ICASSP*, pp. 6419–6423 (cit. on pp. 15, 28, 31).

Lubba, C. H., S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones (2019). “catch22: CAnonical Time-series CHaracteristics”. In: *Data Mining and Knowledge Discovery* (cit. on pp. 11, 40, 49, 50).

Mahmoud, I. B., E. Sarkar, M. Manser, and M. Magimai.-Doss (2024). “Feature Representations for Automatic Meerkat Vocalization Classification”. In: *4th International Workshop on Vocal Interactivity In-and-between Humans, Animals and Robots (VIHAR2024)* (cit. on p. 56).

McCowan, B., S. F. Hanser, and L. R. Doyle (1999). “Quantitative tools for comparing animal communication systems: information theory applied to bottlenose dolphin whistle repertoires”. In: *Animal Behaviour* 57, pp. 409–419 (cit. on p. 82).

Mercado, E. I. and S. Handel (2012). “Understanding the structure of humpback whale songs (L)”. In: *The Journal of the Acoustical Society of America* 132, pp. 2947–2950 (cit. on p. 82).

Mohamed, A. et al. (2022). “Self-Supervised Speech Representation Learning: A Review”. In: *IEEE Journal of Selected Topics in Signal Processing* 16.6, pp. 1179–1210 (cit. on pp. 13, 56).

Moummad, I., N. Farrugia, and R. Serizel (2024). “Self-Supervised Learning for Few-Shot Bird Sound Classification”. In: *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 600–604 (cit. on p. 56).

Muckenhirn, H., V. Abrol, M. Magimai-Doss, and S. Marcel (2019). “Understanding and Visualizing Raw Waveform-Based CNNs”. In: *Proc. of Interspeech*, pp. 2345–2349 (cit. on pp. 12, 40).

Muckenhirn, H., M. Magimai.-Doss, and S. Marcel (2018). “Towards Directly Modeling Raw Speech Signal for Speaker Verification Using CNNS”. In: *Proc. of ICASSP*, pp. 4884–4888 (cit. on pp. 12, 40, 41).

Nallanthighal, V. S., Z. Mostaani, A. Härmä, H. Strik, and M. Magimai.-Doss (2021). “Deep learning architectures for estimating breathing signal and respiratory parameters from speech recordings”. In: *Neural Networks* 141, pp. 211–224 (cit. on p. 41).

Newman JD, G. P. (1992). “Noncategorical vocal communication in primates: the example of common marmoset phee calls”. In: *Nonverbal vocal communication (eds A Manstead, K Oatley)*, pp. 87–101 (cit. on p. 22).

Niizumi, D., D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino (July 2021). “BYOL for Audio: Self-Supervised Learning for General-Purpose Audio Representation”. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE (cit. on p. 50).

Norcross, J. L. and J. D. Newman (1993). "Context and gender-specific differences in the acoustic structure of common marmoset (Callithrix jacchus) phee calls". In: *American journal of primatology* 30(1), pp. 37–54 (cit. on p. 22).

Oikarinen, T. et al. (2018). "Deep Convolutional Network for Animal Sound Classification and Source Attribution using Dual Audio Recordings". In: *The Journal of the Acoustical Society of America* 145.2, pp. 654–662 (cit. on p. 23).

Okano, H., A. Miyawaki, and K. Kasai (May 2015). "Brain/MINDS: brain-mapping project in Japan". In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 370 (cit. on p. 22).

Oord, A. v. d., Y. Li, and O. Vinyals (2018). *Representation Learning with Contrastive Predictive Coding* (cit. on p. 1).

Osmanski, M. S. and X. Wang (2023). "Perceptual specializations for processing species-specific vocalizations in the common marmoset (Callithrix jacchus)". In: *Proceedings of the National Academy of Sciences of the United States of America* 120.24, e2221756120 (cit. on p. 22).

Osmanski, M. S., X. Song, Y. Guo, and X. Wang (2016). "Frequency discrimination in the common marmoset (Callithrix jacchus)". In: *Hearing Research* 341, pp. 1–8 (cit. on p. 48).

Palaz, D., R. Collobert, and M. Magimai-Doss (2013). "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks". In: *Proc. of Interspeech*, pp. 1766–1770 (cit. on p. 40).

Palaz, D., M. Magimai.-Doss, and R. Collobert (Apr. 2019). "End-to-End Acoustic Modeling using Convolutional Neural Networks for HMM-based Automatic Speech Recognition". In: *Speech Communication* 108, pp. 15–32 (cit. on pp. 12, 40, 44).

Paul, A., H. McLendon, V. Rally, J. T. Sakata, and S. C. Woolley (Apr. 2021). "Behavioral discrimination and time-series phenotyping of birdsong performance". In: *PLOS Computational Biology* 17.4, pp. 1–21 (cit. on pp. 11, 39, 49).

Peng, Y., K. Kim, F. Wu, P. Sridhar, and S. Watanabe (2023). "Structured Pruning of Self-Supervised Pre-Trained Models for Speech Recognition and Understanding". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5 (cit. on p. 69).

Pérez-Espinosa, H. et al. (Jan. 2018). "Automatic individual dog recognition based on the acoustic properties of its barks". In: *Journal of Intelligent & Fuzzy Systems* 34.5, pp. 3273–3280 (cit. on p. 58).

Phaniraj, N., K. Wierucka, Y. Zürcher, and J. M. Burkart (2023). "Who is calling? Optimizing source identification from marmoset vocalizations with hierarchical machine learning classifiers". In: *Journal of Royal Society Interface* (cit. on pp. 11, 22, 23, 39, 49).

Pistorio, A. L., B. Vintch, and X. Wang (Sept. 2006). "Acoustic analysis of vocal development in a New World primate, the common marmoset (Callithrix jacchus)". In: *Journal of the Acoustical Society of America* 120.3, pp. 1655–1670 (cit. on p. 53).

Pomberger, T., J. Löschner, and S. R. Hage (2020). "Compensatory mechanisms affect sensorimotor integration during ongoing vocal motor acts in marmoset monkeys". In: *The European journal of neuroscience* 52(6), pp. 3531–3544 (cit. on p. 22).

Pomberger, T., C. Risueno-Segovia, J. Löschner, and S. R. Hage (2018). "Precise Motor Control Enables Rapid Flexibility in Vocal Behavior of Marmoset Monkeys". In: *Current biology* 28(5), pp. 788–794 (cit. on p. 22).

Prather, J. F. (2013). "Auditory signal processing in communication: Perception and performance of vocal sounds". In: *Hearing Research* 305. Communication Sounds and the Brain: New Directions and Perspectives, pp. 144–155 (cit. on p. 2).

Purohit, T., S. Yadav, B. Vlasenko, S. P. Dubagunta, and M. Magimai.-Doss (2023). "Towards Learning Emotion Information from Short Segments of Speech". In: *Proc. of ICASSP*, pp. 1–5 (cit. on p. 41).

Radford, A., J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever (July 2023). "Robust Speech Recognition via Large-Scale Weak Supervision". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 28492–28518 (cit. on p. 1).

Riviere, M., A. Joulin, P.-E. Mazaré, and E. Dupoux (2020). "Unsupervised pretraining transfers well across languages". In: *Proc. of ICASSP*. IEEE, pp. 7414–7418 (cit. on pp. 15, 28, 31).

Rosenblatt, F. (1957). "The Perceptron: A Perceiving and Recognizing Automaton". In: *Technical Report* 85-460-1 (cit. on p. 7).

Roy, S., C. T. Miller, D. Gottsch, and X. Wang (Nov. 2011). "Vocal control by the common marmoset in the presence of interfering noise". In: *Journal of Experimental Biology* 214.21, pp. 3619–3629 (cit. on p. 22).

Ruff, Z. J., D. B. Lesmeister, C. L. Appel, and C. M. Sullivan (2020). "A Convolutional Neural Network and R-Shiny App for Automated Identification and Classification of Animal Sounds". In: *bioRxiv* (cit. on p. 46).

Rukstalis, M. and J. French (Feb. 2005). "Vocal buffering of the stress response: exposure to conspecific vocalizations moderates, urinary cortisol excretion in isolated marmosets". In: *Hormones and behavior* 47, pp. 1–7 (cit. on p. 22).

Saeed, A., D. Grangier, and N. Zeghidour (2021a). "Contrastive Learning of General-Purpose Audio Representations". In: *Proc. of ICASSP*, pp. 3875–3879 (cit. on p. 23).

Saeed, A., D. Grangier, and N. Zeghidour (2021b). "Contrastive Learning of General-Purpose Audio Representations". In: *Proc. of ICASSP*, pp. 3875–3879 (cit. on p. 28).

Sainburg, T., M. Thielk, and T. Q. Gentner (2020). "Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires". In: *PLoS Computational Biology* 16.10, e1008228 (cit. on p. 2).

Sajjad, H., F. Dalvi, N. Durrani, and P. Nakov (2023). "On the effect of dropping layers of pre-trained transformer models". In: *Computer Speech & Language* 77, p. 101429 (cit. on p. 69).

Sarkar, E. and M. Magimai-Doss (July 2025a). "Leveraging Sequential Structure in Animal Vocalizations". In: Idiap-RR-06-2025 (cit. on p. 81).

Sarkar, E. and M. Magimai.-Doss (2023). "Can Self-Supervised Neural Representations Pre-Trained on Human Speech distinguish Animal Callers?" In: *Proc. of Interspeech*, pp. 1189–1193 (cit. on pp. 23, 27, 38, 40, 41, 48, 50, 56, 57, 59, 92).

Sarkar, E. and M. Magimai.-Doss (2024). “On the Utility of Speech and Audio Foundation Models for Marmoset Call Analysis”. In: *4th International Workshop on Vocal Interactivity In-and-between Humans, Animals and Robots (VIHAR2024)* (cit. on pp. 24, 44, 47, 56, 58, 59, 61).

Sarkar, E. and M. Magimai.-Doss (2025b). “Comparing Self-Supervised Learning Models Pre-Trained on Human Speech and Animal Vocalizations for Bioacoustics Processing”. In: *Proc. of ICASSP*, pp. 1–5 (cit. on p. 55).

Sarkar, E., A. Mohammadi, and M. Magimai-Doss (July 2025). “Adaptation of Speech and Bioacoustics Models”. In: Idiap-RR-05-2025 (cit. on p. 65).

Sarkar, E., K. Wierucka, A. B. Bosshard, J. Burkart, and M. Magimai.-Doss (2025). “On Feature Representations for Marmoset Voca Communication Analysis”. In: *Bioacoustics*, pp. 1–15 (cit. on pp. 37, 48, 59).

Sayigh, L. et al. (Feb. 2017). “The Watkins Marine Mammal Sound Database: An online, freely accessible resource”. In: *Proceedings of Meetings on Acoustics* 27.1, p. 040013 (cit. on pp. 25, 57).

Schmidhuber, J. (2015). “Deep learning in neural networks: An overview”. In: *Neural Networks* 61, pp. 85–117 (cit. on p. 1).

Sethi, S. S. (2020). “Automated acoustic monitoring of ecosystems”. PhD thesis. Imperial College London, UK (cit. on pp. 11, 39, 49).

Seyfarth, R. and D. Cheney (Feb. 2003). “Signalers and Receivers in Animal Communication”. In: *Annual review of psychology* 54, pp. 145–73 (cit. on p. 22).

Seyfarth, R. M. and D. L. Cheney (2010). “Production, usage, and comprehension in animal vocalizations”. In: *Brain and Language* 115, pp. 92–100 (cit. on p. 1).

Shen, J. et al. (2018). “Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4779–4783 (cit. on p. 2).

Shi, R., K. Itoyama, and K. Nakadai1 (2024). “Bird Vocalization Embedding Extraction Using Self-Supervised Disentangled Representation Learning”. In: *Proc. of 4th International Workshop on Vocal Interactivity in-and-between Humans, Animals and Robots (VIHAR)* (cit. on p. 56).

Snowdon, C. T. and A. M. Elowson (2001). “’Babbling’ in pygmy marmosets: Development after infancy”. In: *Behaviour* 138(10), pp. 1235–1248 (cit. on p. 24).

Snyder, D., D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur (2018a). “X-Vectors: Robust DNN Embeddings for Speaker Recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333 (cit. on p. 30).

Snyder, D., D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur (2018b). “X-vectors: Robust DNN embeddings for speaker recognition”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333 (cit. on p. 1).

Stowell, D. (2022a). “Computational bioacoustics with deep learning: a review and roadmap”. In: *PeerJ* 10, e13152 (cit. on pp. 27, 28).

Stowell, D. (2022b). “Computational bioacoustics with deep learning: a review and roadmap”. In: *PeerJ* 10, e13152 (cit. on p. 56).

Stowell, D., T. Petrusková, M. Šálek, and P. Linhart (2019). "Automatic acoustic identification of individuals in multiple species". In: *Methods in Ecology and Evolution* 10.7, pp. 965–976 (cit. on p. 2).

Takahashi, D., A. Fenley, and A. Ghazanfar (May 2016). "Early development of turn-taking with parents shapes vocal acoustics in infant marmoset monkeys". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 371, p. 20150370 (cit. on p. 22).

Takahashi, D. Y., D. Z. Narayanan, and A. A. Ghazanfar (2013). "Coupled Oscillator Dynamics of Vocal Turn-Taking in Monkeys". In: *Current Biology* 23.21, pp. 2162–2168 (cit. on p. 30).

Trigeorgis, G. et al. (2016). "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network". In: *Proc. of ICASSP*, pp. 5200–5204 (cit. on p. 40).

Turesson, H. K., S. Ribeiro, D. R. Pereira, J. P. Papa, and V. H. C. de Albuquerque (Sept. 2016). "Machine Learning Algorithms for Automatic Classification of Marmoset Vocalizations". In: *PLOS ONE* 11, pp. 1–14 (cit. on p. 23).

van den Oord, A. et al. (2016). "WaveNet: A Generative Model for Raw Audio". In: *9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, p. 125 (cit. on p. 2).

Vapnik, V. N. and A. Y. Lerner (1963). "Recognition of Patterns with help of Generalized Portraits". In: *Avtomat. i Telemekh.* 24.6, pp. 774–780 (cit. on p. 16).

Vaswani, A. et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. (cit. on p. 1).

Verma, S. et al. (2017). "Discovering Language in Marmoset Vocalization". In: *Proc. Interspeech 2017*, pp. 2426–2430 (cit. on p. 23).

Wierucka, K. et al. (2024). "Same data, different results? Evaluating machine learning approaches for individual identification in animal vocalisations". In: *bioRxiv* (cit. on p. 23).

Wisler, A., L. J. Brattain, R. Landman, and T. F. Quatieri (2016). "A Framework for Automated Marmoset Vocalization Detection and Classification". In: *Proc. Interspeech 2016*, pp. 2592–2596 (cit. on p. 23).

Worley, K. and al. (July 2014). "The common marmoset genome provides insight into primate biology and evolution". In: *Nature Genetics* Nat Genet. 2014 Aug;46(8):850-7. Pp. 850–857 (cit. on p. 22).

Wu, H.-H. et al. (2021). "Multi-Task Self-Supervised Pre-Training for Music Classification". In: *Proc. of ICASSP*, pp. 556–560 (cit. on p. 28).

Yang, S.-w. et al. (2021). "SUPERB: Speech Processing Universal PERformance Benchmark". In: *Proc. of Interspeech*, pp. 1194–1198 (cit. on pp. 28, 30, 41, 50, 59).

Zazo, R., T. N. Sainath, G. Simko, and C. Parada (2016). "Feature Learning with Raw-Waveform CLDNNs for Voice Activity Detection". In: *Proc. of Interspeech*, pp. 3668–3672 (cit. on p. 40).

Zeng, M., X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu (2021). "MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, pp. 791–800 (cit. on p. 28).

Zhang, Y.-J., J. Huang, N. Gong, Z.-H. Ling, and Y. Hu (July 2018). "Automatic detection and classification of marmoset vocalizations using deep and recurrent neural networks". In: *The Journal of the Acoustical Society of America* 144, pp. 478–487 (cit. on pp. 2, 29).

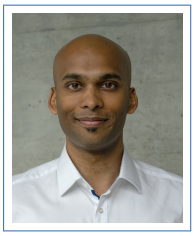
Zhang, Y., J. Huang, N. Gong, Z. Ling, and Y. Hu (July 2018). “Automatic detection and classification of marmoset vocalizations using deep and recurrent neural networks”. In: *The Journal of the Acoustical Society of America* 144, pp. 478–487 (cit. on p. 23).

Zürcher, Y. and J. M. Burkart (2017). “Evidence for dialects in three captive populations of common marmosets (Callithrix jacchus)”. In: *International Journal of Primatology* 38.4, pp. 780–793 (cit. on p. 22).

Eklavya Sarkar

PhD Candidate

Rue de la Poste 1
1920 Martigny, CH
+41 78 82 50 754
eklavya.sarkar@idiap.ch
https://eklavyafcb.github.io
Nationality: Swiss



Research Interests

Speech Processing, Computer Vision, Self-Supervised Learning, Deep Learning

Work Experience

2021–Present **Research Assistant (PhD)**, *Idiap Research Institute*, EPFL, CH
Supervisor: Dr. Mathew Magimai Doss, Senior Researcher, Speech and Audio Processing.

- Led interdisciplinary research bridging deep learning, speech processing, and bioacoustics, with a focus on transferability of speech models for decoding non-human communication.
- Published multiple first-author, peer-reviewed papers in top machine learning conferences and journals, including ICASSP and Interspeech, accumulating 120+ citations.
- *Research areas*: Self-supervised learning, foundation models, fine-tuning, audio processing, bioacoustics, voice activity detection, call-type classification, discrete acoustic tokens.

2020 – 21 **Research Intern**, *Idiap Research Institute*, Martigny, CH
Supervisor: Dr. Sébastien Marcel, Senior Researcher, Biometrics Security and Privacy.

- Designed and implemented novel generative models by adding losses to StyleGAN2.
- Investigated vulnerabilities of modern facial recognition systems against deepfake attacks.

2017 **Research Intern**, *CERN*, Geneva, CH
Supervisor: Dr. Archana Sharma, Principal Scientist, CMS Experiment.

- Refined production code efficiency by completing pull requests on data acquisition tools.
- Contributed to open-source data acquisition tools and radiation physics R&D experiments.

Education

2021–Present **PhD Machine Learning**, *Ecole Polytechnique Fédérale de Lausanne*, CH, (5.2/6.0).

2018 – 19 **MSc Data Science**, *University of Bath*, UK, Distinction.

2015 – 18 **BSc Computer Science**, *University of Liverpool*, UK, Distinction.

Publications

ICASSP 2025 **Sarkar, E.**, Magimai-Doss, M., *Comparing Self-Supervised Learning Models Pre-Trained on Human Speech and Animal Vocalizations for Bioacoustics Processing.*

Bioacoustics 2025 **Sarkar, E.**, K. Wierucka, A. B. Bosshard, J. M. Burkart, Magimai-Doss, M., *On Feature Representation for Marmoset Vocal Communication Analysis.*

Interspeech 2024 **Sarkar, E.**, Magimai-Doss, M., *On the utility of Speech and Audio Foundation Models for Animal Call Analysis.*, 4th International Workshop on VIHAR.

Interspeech 2024 Ben Mahmoud, I., **Sarkar, E.**, Manser, M., Magimai-Doss, M., *Feature Representations for Automatic Meerkat Vocalization Classification.*, 4th International Workshop on VIHAR.

Interspeech 2023 **Sarkar, E.**, Magimai-Doss, M., *Can Self-Supervised Neural Representations Pre-Trained on Human Speech distinguish Animal Callers?*

Interspeech 2022 **Sarkar, E.**, Prasad, R., Magimai-Doss, M., *Unsupervised Voice Activity Detection by Modeling Source and System Information using Zero Frequency Filtering*.

ICASSP 2022 **Sarkar, E.**, Korshunov, P., Colbois, L. and Marcel, S., *Are GAN-based Morphs Threatening Face Recognition?*

Thesis

Ph.D. Transferability of Learnt Speech Representations for Decoding Non-Human Vocal Communication.

M.Sc. Optimising Facial Information Extraction and Processing using Deep Learning.
Grade: Distinction.

B.Sc. Unsupervised Machine Learning: Kohonen Self-Organizing Maps.
Grade: Distinction (90%).

Academic Projects

RL **Flappy Bird Deep Q-Learning Network**

- Trained model to play Flappy Bird using a DQN, and surpassed human level performance.
- Refined optimal policy with Experience Replay and Deep Deterministic Policy Gradients.

NLP **Open Information Relation Extraction**

- Summarised body of text by training a ML speech tag classifier using Glove word vectors.
- Improved model by coding backtracking, Viterbi algorithm, Adam optimiser from scratch.

Leadership Experience

2022–24 **Organizer**, *Perspectives on AI Symposium Series*, Idiap Research Institute.

- Participated in organization: finding sponsors, budgeting, designing the event website.

2017–18 **President**, *Students Residence Society*, University of Liverpool.

Academic Duties and Mentorship

Fall '21-24 **Lead Teaching Assistant**, Master in Artificial Intelligence, UniDistance Suisse.

- Led TAs to grade assignments, exams, & provide critical feedback for a 4 ETCS module.

Reviewer, IEEE Signal Processing Letters, IEEE Transactions on Technology and Society.

Awards

Aug 2020 **International Create Challenge**, *3rd Prize*, AI-Hackathon. Adversarial Attacks Detection.

Programming Skills

Languages Python, Java, Javascript, PHP, SQL, C++, C#, TEX, HTML, CSS.

Frameworks Hydra, PyTorch, Lightning, Optuna, Keras, SkLearn, D3.js.

Misc. Git, Unix, W&B, Mamba/Conda, SGE, Jupyter, Kaggle, Colab, xCode, Eclipse.

Languages

Fluent: English, French, Hindi. Intermediate: German.