Towards Leveraging Sequential Structure in Animal Vocalizations

Eklavya Sarkar
Idiap Research Institute
Ecole Polytechnique Fédérale de Lausanne
Switzerland
`eklavya.sarkar@idiap.ch`

Mathew Magimai.-Doss,
Idiap Research Institute
Switzerland
`mathew@idiap.ch`

Abstract

Animal vocalizations contain sequential structures that carry important communicative information, yet most computational bioacoustics studies average the extracted frame-level features across the temporal axis, discarding the order of the sub-units within a vocalization. This paper investigates whether discrete acoustic token sequences, derived through vector quantization and gumbel-softmax vector quantization of extracted self-supervised speech model representations can effectively capture and leverage temporal information. To that end, pairwise distance analysis of token sequences generated from HuBERT embeddings shows that they can discriminate call-types and callers across four bioacoustics datasets. Sequence classification experiments using k-Nearest Neighbour with Levenshtein distance show that the vector-quantized token sequences yield reasonable call-type and caller classification performances, and hold promise as alternative feature representations towards leveraging sequential information in animal vocalizations.

1 Introduction

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 [1]. 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 [2], while cetaceans exhibit intricate, temporally-structured acoustic sequences associated with social interaction and individual identification [3]. Thus, capturing and analyzing the inherent sequential structure in animal vocalizations could substantially enhance our understanding of their communicative function and biological significance.

However, in many existing computational bioacoustics works [4, 5, 6, 7, 8], each sample's extracted feature embeddings $\boldsymbol{x} \in \mathbb{R}^{N\times D}$ are typically averaged into a vocalization-level representation, denoted as a functional vector $f_\mu = \mu(\boldsymbol{x}) \in \mathbb{R}^D$ 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 [9]. The goal of this paper 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.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Non-Human Animal Communication.

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 [10]. Based on this prior, we extend this framework to bioacoustics, and explore whether discretization of animal vocalizations into token sequences can reveal meaningful structure 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 bioacoustics. To that end, we investigate different discretization methods, namely vector quantization and gumbel-softmax vector quantization, through a distance analysis, and then benchmark the downstream classification performance to evaluate their efficiency towards leveraging the sequential structure in animal calls.

The rest of this paper is organized as follows: Section 2 presents a review of representation learning using discrete audio tokens, Section 3 describes our experimental setup, and Section 4 presents the results and analysis. Finally, Section 5 concludes the paper with directions for future work.

2 Discrete Audio Tokens-based Representation Learning

Modern self-supervised learning (SSL) foundation models pre-trained on human speech have shown strong transferability to bioacoustic tasks [11]. Many such SSL models employ discrete token representations during their pre-training stages which are typically derived using a discretization process, either through integrated vector quantization layers [12, 13] or offline clustering mechanisms applied to continuous embeddings [14]. However, such discrete representations are primarily intended to facilitate SSL objectives, and are usually not directly exposed or utilized during inference or downstream tasks. In this paper, we explicitly leverage the discrete tokenization methodology. To do so, we first extract window-level embeddings from a pre-trained SSL model, 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 would allow the codebook vectors to adapt specifically to the acoustic characteristics and distributions of the calls being studied.

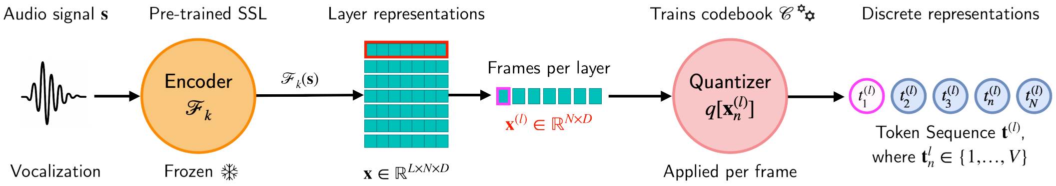

Figure 1: Discrete call tokenization pipeline using vector quantization.

The overall call tokenization pipeline, employed in this work, using vector quantization is illustrated in Figure 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 $q : \mathbb{R}^D \to \{1, 2, \dots, V\}$ 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 $\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_V\}$, where each code-vector $\boldsymbol{c}_i \in \mathbb{R}^D$ corresponds to the i-th discrete token in the original embedding space.

3 Experimental Setup

Common marmosets (*Callithrix jacchus*) are highly vocal new world primates with an extensive vocal repertoire, often studied to understand the evolutionary origins of human vocal communication. For our study, we conducted our experiments on the three marmosets datasets from [7], namely the

InfantMarmosetsVox (IMV) [4], *Bosshard* [15, 16], and *Wierucka* datasets. We also included the dogs dataset [17], referred to as *Abzaliev*. For further dataset details, the reader is referred to Appendix B. To evaluate performance, we look at the call-type (CTID) and caller (CLID) classification tasks. Given the demonstrated transferability of learnt speech representations to animal vocalizations [11], we select HuBERT as our pre-trained SSL model based on its existing benchmarking and effectiveness on bioacoustics tasks [18], and first extract embeddings $\boldsymbol{x} \in \mathbb{R}^{L \times N \times D}$ from the raw input calls.

To discretize the animals calls we investigate vector quantization (VQ) and gumbel-softmax vector quantization (GVQ) due to their proven effectiveness in quantizing audio embeddings. Their respective implementations are detailed in Appendix A.1 and A.2. We first train them on $\boldsymbol{x}$ based on our established protocols (A.3), and then proceed to generate and post-process token sequences $\boldsymbol{t}$ for all data samples (A.4). Note that most modern acoustic tokenizers have multiple quantizers, however, for simplicity and clarity, we focus only on hand-coded single-codebook ones in this work.

As our first investigation step, we examine the intra and inter-class distances between all the generated token sequences to assess their degree of separability. We compute all the pairwise Levenshtein distances, categorizing comparisons into: (i●) *Intra-caller, intra-calltype*: two vocalizations from the same caller producing the same call-type (smallest expected distance). (ii●) *Intra-caller, inter-calltype*: two vocalizations from the same caller producing different call-types. (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 (largest expected distance).

Based on the insights of the comparative analysis, we then evaluate how well the sequential nature of token representations can be leveraged for CTID and CLID. We classify the token sequences with the k-NN algorithm, using the Levenshtein distance as the similarity metric. The predicted label of a sample is determined by applying a majority-voting framework on the actual labels of the k most similar sequences. The classifier training and hyperparameter details are given in D. Finally, we evaluate this tokenization approach to a 'traditional' linear-probing baseline, i.e. by pooling the temporal information into a functional $\boldsymbol{f}_{\sigma\mu} \in \mathbb{R}^{2D}$, and classifying it with a fully-connected layer.

4 Results

4.1 Distance Analysis

Figure 2 presents the means of the distances distributions of the four aforementioned categories. For the token sequences generated from the VQ model (top row) 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.

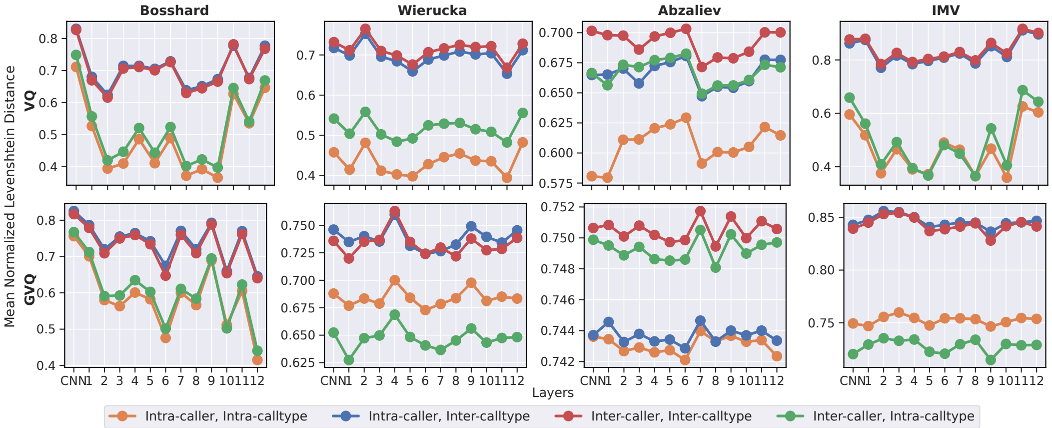

Figure 2: Layer-wise mean Levenshtein distance between all pairs of VQ and GVQ token sequences.

While we can observe similar trends with the GVQ tokens (bottom row) on Bosshard, the remaining datasets exhibit some different patterns. Notably, group (ii ●) and (iii ●)'s trends are flipped in Abzaliev. 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 produced by different callers than those of different calls produced by the same caller. Additionally, for Wierucka and IMV, (iii ●) is unexpectedly smaller on average than (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.

This analysis shows that VQ tokens are indeed capable of clustering sufficient acoustic information to discriminate calls or callers, under real-world left-to-right temporal constraints. The GVQ tokens exhibit some unexpected patterns and less consistent separability, indicating they may be less effective.

4.2 Classification Analysis

The classification results are shown in Figure 3 using the VQ and GVQ tokens. We can observe that the linear baseline clearly outperforms the k-NN classification of token sequences in all scenarios, showing that the HuBERT feature embeddings currently capture more meaningful information. GVQ notably underperforms on all tasks and datasets, especially Abzaliev, essentially achieving chance-level performance, strongly suggesting a codebook collapse to a restricted subset of symbols. On the other hand, VQ yields a weaker performance than the linear baseline, but still achieves reasonable results in many scenarios. This indicates that some degree of call or caller-level discrimination can already be captured simply with the Levenshtein distance, as also seen in Section 4.1, but substantial improvements in classification performance could potentially be achieved through more sophisticated sequence modeling of the generated token sequences. Finally, while a single shared codebook can still encode enough information for call discrimination, it is perhaps not expressive enough to preserve the finer caller-specific nuances existing in the continuous embeddings.

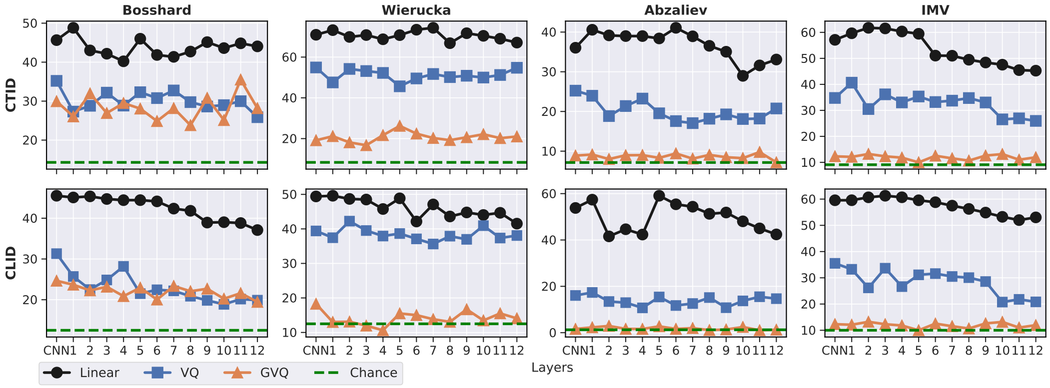

Figure 3: Layer-wise UAR [%] for CTID (top) and CLID (bottom) using k-NN on token sequences.

Table 1 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 4 visually plots the same information. For CTID, discretizing the feature embeddings with a VQ and GVQ drops the performance across datasets by $\sim$26-39% and $\sim$27-79% respectively, when compared to stats-pooling the same features and then classifying with a linear layer. For CLID, the drop is of $\sim$15-71% and $\sim$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 could thus be to train a quantizer model which employs *multiple* codebooks to retain a richer set of temporal patterns.

Table 1: 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

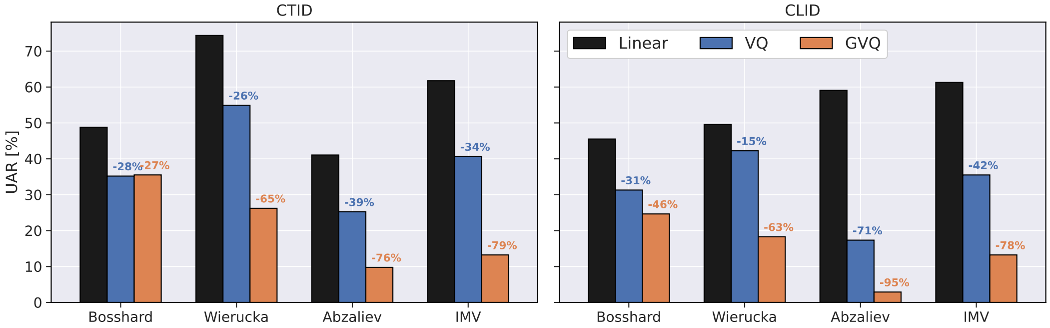

Figure 4: Best UAR results across layers for CTID and CLID.

Lastly, although we trained a single codebook, shared across all layers, for both VQ and GVQ, we can still observe in Figure 3 that earlier layers tend to yield better performance across tasks, consistent with the trends reported in other works [7, 18]. 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.

5 Conclusions and Future Work

In this paper, we explored alternate feature representations that could preserve the temporal structure of animal vocalizations instead of averaging their extracted feature embeddings into single functional vectors. To that end, we investigated call discretization and evaluated whether discrete acoustic token modeling could effectively improve animal call classification performance. We first trained a vector quantization and a gumbel-softmax vector quantization module to convert vocalizations into discrete token sequences for four different animal datasets. A distance analysis of generated sequences showed that they are able to encode temporal information, and exhibit a degree of separability by call-type or caller identity across all datasets. Training a k-NN classifier on said representations showed that while VQ token representations are still weaker than linear-probing baselines, they are nonetheless able to leverage meaningful sequential information from animal vocalizations, and incorporating more sophisticated sequence modeling could further improve performance.

The scope for improvements on this topic is fairly substantial. One line of future work could explore larger, multi-codebook quantization architectures, such as Residual VQs [19] or Grouped VQ [20]. Another direction could investigate other sequence post-processing techniques, such as deduplication, i.e. removing consecutive duplicate tokens [21], or acoustic byte-pair encoding (BPE) [22], which can further reduce the sequence length and tighten the alignment between tokens and acoustically meaningful sub-units.

Acknowledgments and Disclosure of Funding

This work was funded by the Swiss National Science Foundation's NCCR Evolving Language project (grant no. 51NF40_180888).

References

[1] Arik Kershenbaum, Daniel T. Blumstein, Marie A. Roch, Çaglar Akçay, George Backus, Mark A. Bee, Kirsten Bohn, Yan Cao, Gerald Carter, Cristiane Cäsar, Michael Coen, Stacy Lynn DeRuiter, Laurance Doyle, Shimon Edelman, Ramon Ferrer-i Cancho, Todd M. Freeberg, Ellen Clare Garland, Morgan Gustison, Heidi E. Harley, Chantal Huetz, Melissa Hughes, Janelle Hyland Bruno, Amiyaal Ilany, Dezhe Z. Jin, Michael Johnson, Chenghui Ju, James Karnowski, Bernard Lohr, Marta B. Manser, Brenda McCowan, Eduardo Mercado, Peter M. Narins, Alex Piel, Meg Rice, Roberta Salmi, Kazutoshi Sasahara, Laela Sayigh, Yu Shiu, Charles Taylor, Elena E. Vallejo, Sara Waller, and Veronica Zamora-Gutierrez. Acoustic sequences in non-human animals: a tutorial review and prospectus. *Biological Reviews*, 91(1):13–52, 2016.

[2] C. K. Catchpole and P. J. B. Slater. *Bird Song: Biological Themes and Variations*. Cambridge University Press, Cambridge, 2003.

[3] E. I. Mercado and S. Handel. Understanding the structure of humpback whale songs (l). *The Journal of the Acoustical Society of America*, 132:2947–2950, 2012.

[4] Eklavya Sarkar and Mathew Magimai.-Doss. Can self-supervised neural representations pre-trained on human speech distinguish animal callers? In *Proc. of Interspeech*, pages 1189–1193, 2023.

[5] Eklavya Sarkar and Mathew Magimai.-Doss. 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)*, 2024.

[6] Imen Ben Mahmoud, Eklavya Sarkar, Marta Manser, and Mathew Magimai.-Doss. Feature representations for automatic meerkat vocalization classification. In *4th International Workshop on Vocal Interactivity In-and-between Humans, Animals and Robots (VIHAR2024)*, 2024.

[7] Eklavya Sarkar, Kaja Wierucka, Alexandra B. Bosshard, Judith Burkart, and Mathew Magimai.-Doss. On feature representations for marmoset voca communication analysis. *Bioacoustics*, pages 1–15, 2025.

[8] Eklavya Sarkar, Amir Mohammadi, and Mathew Magimai-Doss. Adaptation of speech and bioacoustics models. *Idiap-RR*, (Idiap-RR-05-2025), July 2025.

[9] A. Kershenbaum, D.T. Blumstein, M.A. Roch, Ç. Akçay, G. Backus, M.A. Bee, K. Bohn, Y. Cao, G. Carter, C. Cäsar, M. Coen, S.L. DeRuiter, L. Doyle, S. Edelman, R. Ferrer-i Cancho, T.M. Freeberg, E.C. Garland, M. Gustison, H.E. Harley, C. Huetz, M. Hughes, J. Hyland Bruno, A. Ilany, D.Z. Jin, M. Johnson, C. Ju, J. Karnowski, B. Lohr, M.B. Manser, B. McCowan, E. Mercado, P.M. III, Narins, A. Piel, M. Rice, R. Salmi, K. Sasahara, L. Sayigh, Y. Shiu, C. Taylor, E.E. Vallejo, S. Waller, and V. Zamora-Gutierrez. Acoustic sequences in non-human animals: a tutorial review and prospectus. *Biological Reviews*, 91(1):13–52, 2016.

[10] Yiwei Guo, Zhihan Li, Hankun Wang, Bohan Li, Chongtian Shao, Hanglei Zhang, Chenpeng Du, Xie Chen, Shujie Liu, and Kai Yu. Recent advances in discrete speech tokens: A review, 2025.

[11] Eklavya Sarkar. *Transferability of Learnt Speech Representations for Decoding Non-Human Vocal Communication*. PhD thesis, EPFL, Lausanne, 2025.

[12] Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. In *International Conference on Learning Representations*, 2020.

[13] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

[14] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[15] A. B. Bosshard. Sequential dynamics in common marmoset vocal strings. Master's thesis, University of Zurich, 2020.

[16] Alexandra B. Bosshard, Judith M. Burkart, Paola Merlo, Chundra Cathcart, Simon W. Townsend, and Balthasar Bickel. Beyond bigrams: call sequencing in the common marmoset (<i>callithrix jacchus</i>) vocal system. *Royal Society Open Science*, 11(11):240218, 2024.

[17] Artem Abzaliev, Humberto Perez-Espinosa, and Rada Mihalcea. Towards dog bark decoding: Leveraging human speech processing for automated bark classification. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, Torino, Italia, May 2024. ELRA and ICCL.

[18] Eklavya Sarkar and Mathew Magimai.-Doss. Comparing self-supervised learning models pre-trained on human speech and animal vocalizations for bioacoustics processing. In *Proc. of ICASSP*, pages 1–5, 2025.

[19] Biing-Hwang Juang and A. Gray. Multiple stage vector quantization for speech coding. In *ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pages 597–600, 1982.

[20] Herve Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

[21] Xuankai Chang, Brian Yan, Kwanghee Choi, Jee-Weon Jung, Yichen Lu, Soumi Maiti, Roshan Sharma, Jiatong Shi, Jinchuan Tian, Shinji Watanabe, Yuya Fujita, Takashi Maekaku, Pengcheng Guo, Yao-Fei Cheng, Pavel Denisov, Kohei Saijo, and Hsiu-Hsuan Wang. 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)*, pages 11481–11485, 2024.

[22] Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, February 1994.

[23] Aaron van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017.

[24] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.

[25] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[26] Y.J. Zhang, J. Huang, N. Gong, Z.H. Ling, and Y. Hu. Automatic detection and classification of marmoset vocalizations using deep and recurrent neural networks. *The Journal of the Acoustical Society of America*, 144:478–487, 07 2018.

[27] J. A. Agamaite, C. J. Chang, M. S. Osmanski, and X. Wang. A quantitative acoustic analysis of the vocal repertoire of the common marmoset (callithrix jacchus). *The Journal of the Acoustical Society of America*, 138(5):2906–2928, 2015.

[28] C. T. Snowdon and A. M. Elowson. 'babbling' in pygmy marmosets: Development after infancy. *Behaviour*, 138(10):1235–1248, 2001.

A Quantization

A.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 [23], 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:

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

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 1, a *straight-through estimator* (STE) [24] 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:

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

where $\text{sg}[\cdot]$ 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 [23], 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.

A.2 Gumbel-Softmax Vector Quantization (GVQ)

To mitigate codebook collapse in the standard VQ, we also implement gumbel vector quantization (GVQ) [25], 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:

$$p_i = \frac{\exp\big((\log \pi_i + g_i)/\tau\big)}{\sum_{j=1}^{V} \exp\big((\log \pi_j + g_j)/\tau\big)}, \tag{3}$$

where each g_i is an independent sample from the $\text{Gumbel}(0, 1)$ 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:

$$\mathcal{L}_{\mathrm{GVQ}} = \sum_{i=1}^{V} p_i \log\bigl(p_i\, V\bigr) \tag{4}$$

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

$$\mathrm{PPL} \;=\; \exp\Bigl(-\sum_{i=1}^{V} \bar{p}_i \log \bar{p}_i\Bigr), \tag{5}$$

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.

A.3 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 2.

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.

Table 2: 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

A.4 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 2. 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.

B Datasets

Table 3 presents a statistical summary of the datasets used in this work. Marmosets are a central focus of this paper, as their vocal behaviour provides a particularly valuable surrogate model for studying the evolutionary origins of human language. Indeed, their relevance to comparative communication science makes them especially well-suited for exploring how vocal signals encode socially and biologically meaningful information across species.

Table 3: 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	$1,037$	$1,687$
Abzaliev	Dogs	$8,034$	137	48	14	80	2	655	1313

InfantMarmosetsVox (IMV) [4] is an extended version of the dataset used in the study on marmoset call type discrimination by [26]. 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 [5]. The calltypes are entitled peep (pre-phee), phee, twitter, trill, trillphee, tsik tse, egg, pheecry (cry), trllTwitter, pheetwitter, and peep calls.

The **Bosshard** [15, 16] 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 [27, 28]. 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.

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 [17]. 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.

C Levenshtein Distance

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.

D Classifier Experimental Setup

We train the k-NN classifier by iterating over the hyperparameters given in Table 4 for each layer to obtain optimal classification results. The classifier is trained on *Train*, and the hyperparameters defined in the search space are evaluted on *Val*, using UAR as the optimization criterion. The best hyperparameters are used on *Test*, and 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 4: 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]

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, the paper clearly shows the limitations of the proposed approach, but discusses potential for improvements.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: All results are justified, but there are no proofs needed or applicable to this applied work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper includes detailed experimental setup, publicly available datasets, and in-depth description of proposed approaches.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The paper uses existing open-source datasets, as well as those available upon request. It also includes detailed description on the process to reproduce results. However, it does not include any source code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All experimental details are given in the main text or in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The paper does not give error bars, but has clear training/val/test splits, as well as details on other optimization parameters. The experiments, however, were not running multiple times with an average and error bars for each result.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The compute resources are relatively minor for a machine learning paper. They just involve feature extraction and classification. These do not require strong CPU/GPU resources, especially since bioacoustics datasets are small.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The researech conducted in this paper fully respects the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the impact of this work for the bioacoustics community.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no real risk of misuse of the work conducted in this paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The authors are original owners of one dataset, and the others are appropriately credited/cited. No other real assets were required or used in this work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: This paper does not release any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research did not involve any LLMs as any important, original, or non-standard components. The research and proposed methods fully originate from the authors.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.