

Can Self-Supervised Neural Representations Pre-Trained on Human Speech distinguish Animal Callers?

Eklavya Sarkar^{1,2} and Mathew Magimai.-Doss¹

¹Idiap Research Institute, Martigny, Switzerland

²Ecole polytechnique fédérale de Lausanne, Switzerland

{eklavya.sarkar, mathew}@idiap.ch

Abstract

Self-supervised learning (SSL) models use only the intrinsic structure of a given signal, independent of its acoustic domain, to extract essential information from the input to an embedding space. This implies that the utility of such representations is not limited to modeling human speech alone. Building on this understanding, this paper explores the cross-transferability of SSL neural representations learned from human speech to analyze bio-acoustic signals. We conduct a caller discrimination analysis and a caller detection study on Marmoset vocalizations using eleven SSL models pre-trained with various pretext tasks. The results show that the embedding spaces carry meaningful caller information and can successfully distinguish the individual identities of Marmoset callers without fine-tuning. This demonstrates that representations pre-trained on human speech can be effectively applied to the bio-acoustics domain, providing valuable insights for future investigations in this field.

Index Terms bio-acoustics, self-supervised learning, caller discrimination and detection, representation learning.

1. Introduction

The study of animal vocalizations, or bio-acoustics, has progressed significantly in recent years due to approaches inherited from machine learning and deep learning [1]. However, most of these are supervised approaches, which require large amounts of labeled data, which is often scarce in bio-acoustics. 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 [2, 3, 4], or predict future frames using an auto-regressive framework [5, 6]. Discriminative approaches either learn by contrastive learning, i.e. discriminating positive samples from negative ones [7, 8], or else by predicting pseudo-labels of discrete masked regions [9, 10, 11] or the output of specific hidden layers [12]. 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 [13].

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 [14, 15] and biomedical signals [16, 17]. 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 bio-acoustics and a much harder problem than across-species classification [1]. Some previous works have explored birdsong detection [18] and bio-acoustic event detection [19] 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.

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.

2.1. Dataset

For our study, we requested and used the marmoset dataset collected and labeled by [20]. The dataset 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 labelled 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 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 final performance of the trained models on unseen data.

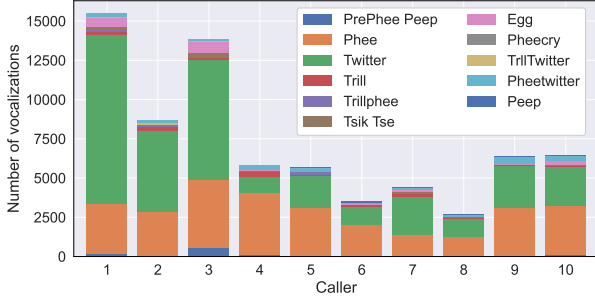


Figure 1: Vocalization per callers grouped by call-type.

2.2. Caller-Groups

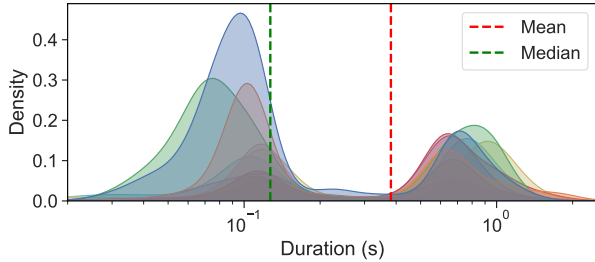


Figure 2: Log distribution of vocalization lengths for callers 1–10 represented in different colors. The mean and median are calculated over the entire dataset.

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 2, exhibits a bimodal structure consistent with prior findings [21, 22]. 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 [23, 24]. 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.

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 [13] based on the different pretext tasks seen in Section 1, and use the S3PRL toolkit [13] to extract the embeddings. Table 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 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 [5]	LS 360	4.11	512	Autoreg. Rec.
VQ-APC [6]	LS 360	4.63	512	Autoreg. Rec.
NPC [2]	LS 360	19.38	512	Masked Rec.
Mockingjay [3]	LS 100	21.33	768	Masked Rec.
TERA [4]	LS 100	21.33	768	Masked Rec.
Mod-CPC [7]	LL 60k	1.84	256	Contrastive
Wav2Vec2 [8]	LS 960	95.04	768	Contrastive
Hubert [9]	LS 960	94.68	768	Masked Pred.
DistilHubert [12]	LS 960	27.03	768	Masked Pred.
WavLM [10]	LS 960	94.38	768	Masked Pred.
Data2Vec [11]	LS 960	93.16	768	Masked Pred.

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 $\mathcal{N}(\mu, \Sigma)$ with mean μ and diagonal covariance matrix Σ , 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. Specifically, for inter-caller distances, we calculate a total of $100 \cdot 100$ pairwise distances for each pair of callers. For inter-caller distances, we compute a total of $\binom{100}{2}$ distances. To compute the distance between 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 $[0, +\infty)$. It is to be noted that the latter provides a symmetric measure while the former does not.

Equations 1 and 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

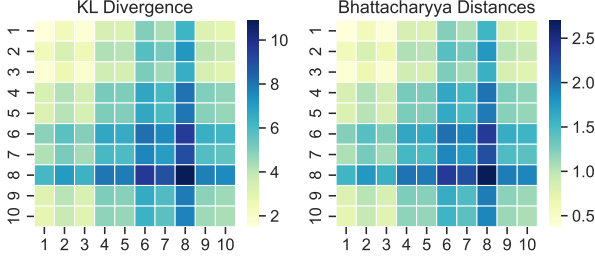


Figure 3: Distance matrix of callers in WavLM’s embedding space. The off-diagonal values represent the average inter-caller distances, while the diagonal entries the average intra-caller distances. Darker regions indicate higher dissimilarity.

\mathcal{N}_g [25, 26]. In the case of the KL divergence, the mean vector μ , covariance matrix Σ , determinant $|\Sigma|$, and dimensionality d are utilized. Meanwhile, the Bhattacharyya distance uses the arithmetic mean of the covariance matrices Σ_f and Σ_g as Σ .

$$D_{KL}(f||g) = \frac{1}{2} \left(\log \frac{|\Sigma_g|}{|\Sigma_f|} + \text{Tr}(\Sigma_g^{-1} \Sigma_f) + (\mu_f - \mu_g)^T \Sigma_g^{-1} (\mu_f - \mu_g) - d \right) \quad (1)$$

$$D_{BC}(f||g) = \frac{1}{8} (\mu_f - \mu_g)^T \Sigma^{-1} (\mu_f - \mu_g) + \frac{1}{2} \log \left(\frac{|\Sigma|}{\sqrt{|\Sigma_f| |\Sigma_g|}} \right) \quad (2)$$

Once we have computed the distribution of distances for all the SSL embedding spaces, we can visualize them through a heatmap. Figure 3 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.

4. Caller Detection Study

4.1. Classifiers

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

Table 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']

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 2.

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.

4.3. Results and Discussion

Table 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 3 summarizes the performance of the different classifiers on all the embedding spaces. The results show that SVM

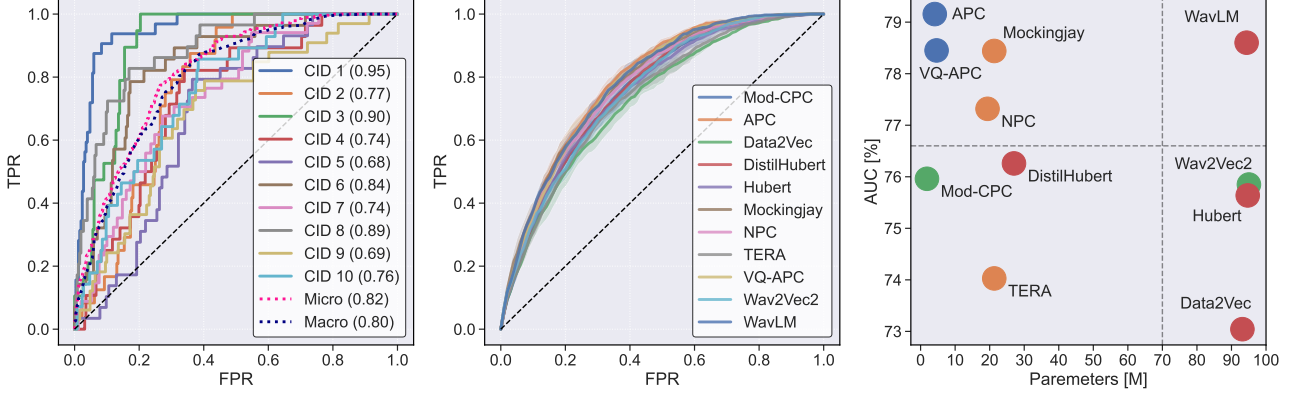


Figure 4: a) ROC curves per caller class (CID) for WavLM embeddings using SVM on one fold of Test. b) Macro average ROC curves of all models on Test using SVM over all folds. Shaded areas represent ± 1 std over the k -folds. c) Model size against performance. Model pre-training objective denoted as: ● Masked prediction. ● Autoregressive reconstruction. ● Contrastive ● Masked reconstruction.

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 4a) 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 4b) 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 4c) 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.

5. Conclusions

This paper investigated the applicability of self-supervised representations, pre-trained on human speech through different approaches, to analyze vocalizations in the bio-acoustics 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 bio-acoustics domain for tasks such as Marmoset caller detection, even without fine-tuning. These findings can greatly benefit bio-acoustics 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 bio-acoustics 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.

6. Acknowledgments

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

7. References

- [1] D. Stowell, “Computational bioacoustics with deep learning: a review and roadmap,” *PeerJ*, vol. 10, p. e13152, 2022.
- [2] A. H. Liu, Y.-A. Chung, and J. Glass, “Non-Autoregressive Predictive Coding for Learning Speech Representations from Local Dependencies,” in *Proc. of Interspeech*, 2021, pp. 3730–3734.
- [3] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, “Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders,” in *Proc. of ICASSP*, 2020, pp. 6419–6423.
- [4] A. T. Liu, S.-W. Li, and H.-y. Lee, “TERA: Self-Supervised Learning of Transformer Encoder Representation for Speech,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, p. 2351–2366, jul 2021.
- [5] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, “An unsupervised autoregressive model for speech representation learning,” in *Proc. of Interspeech*, 2019.
- [6] Y.-A. Chung, H. Tang, and J. Glass, “Vector-quantized autoregressive predictive coding,” in *Proc. of Interspeech*, 2020.
- [7] M. Riviere, A. Joulin, P.-E. Mazaré, and E. Dupoux, “Unsupervised pretraining transfers well across languages,” in *Proc. of ICASSP*. IEEE, 2020, pp. 7414–7418.
- [8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.
- [9] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [10] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, “WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [11] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, “data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language,” in *Proc. of ICML*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 1298–1312.
- [12] H.-J. Chang, S.-w. Yang, and H.-y. Lee, “DistilHuBERT: Speech representation learning by layer-wise distillation of hidden-unit BERT,” in *Proc. of ICASSP*. IEEE, 2022, pp. 7087–7091.
- [13] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, “SUPERB: Speech Processing Universal PERFORMANCE Benchmark,” in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.
- [14] H.-H. Wu, C.-C. Kao, Q. Tang, M. Sun, B. McFee, J. P. Bello, and C. Wang, “Multi-Task Self-Supervised Pre-Training for Music Classification,” in *Proc. of ICASSP*, 2021, pp. 556–560.
- [15] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, “MusicBERT: Symbolic music understanding with large-scale pre-training,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 2021, pp. 791–800.
- [16] H. Banville, O. Chehab, A. Hyvärinen, D.-A. Engemann, and A. Gramfort, “Uncovering the structure of clinical EEG signals with self-supervised learning,” *Journal of Neural Engineering*, vol. 18, no. 4, p. 046020, mar 2021.
- [17] H. Banville, I. Albuquerque, A. Hyvärinen, G. Moffat, D.-A. Engemann, and A. Gramfort, “Self-Supervised Representation Learning from Electroencephalography Signals,” in *International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1–6.
- [18] A. Saeed, D. Grangier, and N. Zeghidour, “Contrastive learning of general-purpose audio representations,” in *Proc. of ICASSP*, 2021, pp. 3875–3879.
- [19] P. C. Bermant, L. Brickson, and A. J. Titus, “Bioacoustic Event Detection with Self-Supervised Contrastive Learning,” *bioRxiv*, 2022.
- [20] 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*, vol. 144, pp. 478–487, 07 2018.
- [21] J. Huang, H. Ma, Y. Sun, L. Chang, and N. Gong, “Complex rules of vocal sequencing in marmoset monkeys,” *bioRxiv*, 2022.
- [22] D. Y. Takahashi, D. Z. Narayanan, and A. A. Ghazanfar, “Coupled oscillator dynamics of vocal turn-taking in monkeys,” *Current Biology*, vol. 23, no. 21, pp. 2162–2168, 2013.
- [23] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-End Factor Analysis for Speaker Verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [24] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-Vectors: Robust DNN Embeddings for Speaker Recognition,” in *Proc. of ICASSP*, 2018, pp. 5329–5333.
- [25] J.-L. Durrieu, J.-P. Thiran, and F. Kelly, “Lower and upper bounds for approximation of the kullback-leibler divergence between gaussian mixture models,” in *Proc. of ICASSP*, 2012.
- [26] A. Bhattacharyya, “On a measure of divergence between two statistical populations defined by their probability distributions,” *Bulletin of the Calcutta Mathematical Society*, vol. 33, pp. 99–109, 1943.