Learning Marmoset Vocal Patterns with a Masked Autoencoder for Robust Call Segmentation, Classification, and Caller Identification

Abstract—**The marmoset, a highly vocal primate, is a key model for studying social-communicative behavior. Unlike human speech, marmoset vocalizations are less structured, highly variable, and recorded in noisy, low-resource conditions. Learning marmoset communication requires joint call segmentation, classification, and caller identification—challenging domain tasks. Previous CNNs handle local patterns but struggle with long-range temporal structure. We applied Transformers using self-attention for global dependencies. However, Transformers show overfitting and instability on small, noisy annotated datasets. To address this, we pretrain Transformers with MAE—a self-supervised method reconstructing masked segments from hundreds of hours of unannotated marmoset recordings. The pretraining improved stability and generalization. Results show MAE-pretrained Transformers outperform CNNs, demonstrating modern self-supervised architectures effectively model low-resource non-human vocal communication.**

Index Terms—**Animal call detection, marmoset vocalization, Transformer, self-supervised learning, MAE, ViT, segmentation, classification, caller identification.**

I. Introduction

The common marmoset (Callithrix Jacchus) is an animal model suitable for studying social vocal communication. First, the marmosets are non-human primates genetically and neurophysiologically close to human [1]. Second, in contrast to such primates as gorillas that primarily use gestures for communication, the marmosets are highly vocal and readily to respond to other marmosets, even non-related or non-pair-bonded ones, particularly when visually hindered such as in forests when vocal contact is crucial for survival [2]. Third, marmosets exhibit human-like conversational turn-taking, exchanging calls resembling coupled oscillators [3], [4]. Fourth, marmosets display prosocial behaviors: similar to human cooperative breeding, marmosets take care of offspring of nonparents [5], [6], reflecting group social communications.

Researchers have been using the marmoset as an animal model to study diseases and mechanisms related to vocal communication comparing with human infant linguistic developments. Uesaka et al. [7] developed a marmoset model of autism disorder — marked by the deficits in social communication, impaired verbal interaction, and verbal perseveration — by feeding pregnant mothers with valproic acid. Using the autism model, they aimed to study the developmental vocal characteristics of autism for early diagnosis and investigate potential medicines to relieve the autism symptoms. The ability of vocal communication is shaped by innate and empirical factors. Researchers manipulate autism-related genes of marmosets [8] to study innate gene-function relationships. Researchers manipulate environments including visual or auditory sensory inputs [9] and parental interaction [10] to study the empirical modification of communicative behaviors.

Studying turn-taking vocal communication between marmosets requires extracting caller and callee information, call contents, and vocal exchanges from the recorded audios. Turesson et al. applied SVM and DNN for marmoset call classification on a small dataset of 321 marmoset calls [11]. Wisler et al. used SVM and decision tree on a larger dataset of 4 call types, each with 400 marmoset calls for classfication [12]. Uesaka et al. employed CNN to classify 3 call types (phee, twitter, and trill) to study the development and autism of the marmosets [7]. However, these studies were limited by either small datasets or focused on only a few call types.

Zhang et al. applied RNN and DNN for segmentation and classification of infant marmoset calls on a dataset that contains 10 call types, each with several thousand calls [13]. Their call types include phee, twitter, trill, trillphee, tsik, ek, pheecry, peep, and two infant-specific categories: ct-trill (twitter-connected-trill) and ct-phee (twitter-connected-phee). Sarkar et al. used the same dataset and implemented self-supervised learning on caller discrimination and classification [14]. These studies have two key limitations. First, both studies [13], [14] treat segmentation as a separate task and assumed known segmentation information when classifying calls or callers. Second, their dataset recorded individual marmosets in isolation, without communicative interaction with other marmosets, thus failing to capture the vocal characteristics of marmosets during social communication.

Oikarinen et al. used a dataset of recordings from paired of pairs marmosets housed together in single cages, enabling close-range vocal interaction [15], [16]. The dataset comprises 36 sessions totally 38 hours, with 8 call types: trill, phee, trillphee, twitter, chirp, tsik, ek, and chatter, along with a noise type that indicates the silences between the calls. Applying a CNN model on the dataset, they achieved segmentation, classification, and caller identification [16].

However, while previous works as [13] and [16] used an RNN or a CNN that maps acoustic segments to call labels, the Transformer structure has been proven to outperform RNN [17] and CNN [18] in both sequential language processing and high-dimensional vision tasks. Transformer utilizes self-attention mechanism that efficiently segregates information parallelly over long distances and captures the global structure of marmoset vocalization more efficiently than RNN and more effectively than CNN. While Transformer is typically constrained by the quadratic complexity of input token length [17], making it challenging to process high-dimensional input spectra that discriminate the marmoset calls such as a phee and a trill. We can address the limitation using Vision Transformer [18] that patchizes the high-dimensional input. We propose to use the Transformer model for segmentation, classification, and caller identification of marmoset vocalizations.

However, Transformer shows the problems of overfitting and unstable training on the target dataset. The available annotated two-stream dataset [15] specifically designed for caller annotation has a limited amount. We found that typical 12-layer Transformer overfits on the dataset just as CNN does. We have to decrease both the number of layers and model dimension to reduce overfitting. The complex task that involves segmentation, classification, and caller identification on noisy data also challenges Transformer training. We observed sudden drops in accuracy, some of which never recover to the original level. Such training instability is more severe when we use larger Transformer models.

To address the problem of the overfitting and unstable training of large Transformer with limited annotated data on challenging tasks, we propose using Masked Autoencoder (MAE) [19] to pretrain the Transformer. We applied MAE on hundreds of hours of marmoset recordings without any annotations. We found that after pretraining, the Transformer shows more stable training and almost no overfitting on the limited dataset. We frozed lower layers to extract MAE features and used higher layers for fine-tuning, enabling us to use a typical 12-layer Transformer without reducing model dimension. The MAE pretrained Transformer works best in practice.

We conducted our experiments on the public dataset [15] with recordings designed to study close-range vocal communication between marmoset pairs who exchange calls close together in a cage.

II. Model

We use a two-stream Transformer model (Figure 1) on the dual-audio recordings with two simultaneously recorded channels that come from two interacting animals. The model employs two stream transformer encoders that process sliding spectral segments from each channel to classify into a target label. The target labels indicate call types and caller identities (e.g., the 'tr2' label denotes a trill call from the second of two animals in interactions). The labels also indicate segmentation information: 'noise' denotes the non-call segment between calls. This approach allows our two-stream Transformer model to perform three crucial tasks:

- **Segmentation**: Identifying the start and end times of each vocalization.
- **Classification**: Determining the type of call (e.g., trill, chirp).

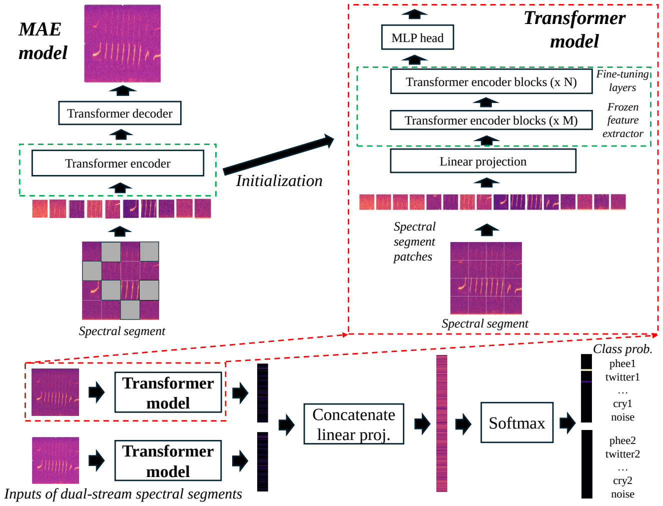


Fig. 1: The two-stream Transformer model. The Transformer model can be initialized with MAE encoder pretrained by spectral segments without labels. We can freeze the first N layers for MAE feature extraction and the later N layers for fine-tuning. Two Transformer models process two spectral segments extracted from a dual-channel audio recording to map to a vocal call label. The call label of 'phee2' denotes that the phee call comes from the second marmoset of the pair in the cage; the 'noise' denotes the non-call segment between calls, used for segmenting calls to non-calls. We used two output heads to enable simultaneous call prediction, allowing detection of concurrent calls of the animal pair such as 'trill1' and 'twitter2'.

- **Caller identification**: Attributing each call to the correct animal.

By integrating these functions into a single model, we capture the complex dynamics of animal interactions through their vocalizations.

For our two-stream Transformer model, we utilize two Vision Transformer [17] modules; each (Figure 1) processes the high-resolution linear spectral image by dividing it into patches that form a sequence of patch tokens. These tokens undergo a linear transformation and are augmented with positional encoding and a learnable class token. The resulting sequence passes through a typical Transformer architecture that comprises alternated self-attention and feedforward modules. The two-stream Transformer's vision transformer modules output two encoded class tokens, which are then linear projected and concatenated, passing through a shared linear layer for final class label prediction.

We can initialize each stream of Vision Transformer with identical parameters from the SSL model of MAE pretrained on spectral segments without labels. We froze the first M layers for MAE feature extraction and the later N layers for task-specific fine-tuning. This method retains the generalizability of MAE features while fine-tuning fewer parameters, making training faster and reducing overfitting risk for small annotated

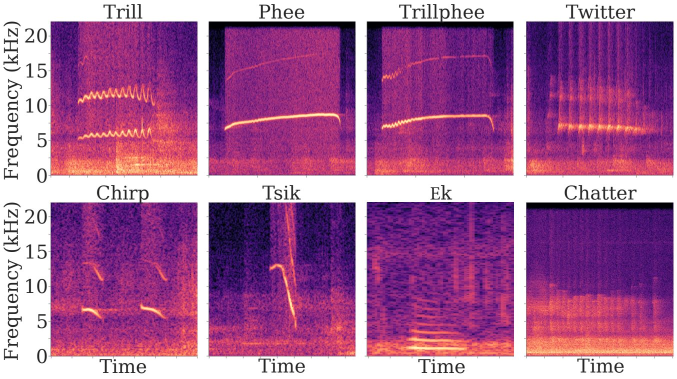


Fig. 2: Spectrograms of marmoset calls. Note that ranges in time axis vary greatly among calls, e.g., 0.01 second for ek and 4 seconds for chatter. Those call examples come from the dataset [15]. We experimented on this dataset with the 8 call types, same as [16].

datasets.

III. Evaluation

We use F-score and accuracy to evaluate classification, segmentation, and caller identification, same as [16].

For caller evaluation, we convert predicted labels into two separate segment files for a marmoset pair (e.g., 'tr2' labels are converted to 'tr' and added to the segment file of the second marmoset).We evaluate two predicted segment files against corresponding annotated ones for marmoset pair housed together.

For segmentation evaluation, we process the annotated and predicted segment files. First, we add 'noise' labels to fill in the intervals without any calls in the annotated segment files. Second, we reconstruct the interval for each call by merging predicted successive identical labels in the predicted segment files (e.g., the predicted 'noise, tr, tr, tr, noise' sequence indicates a three-time-unit trill call surrounded by noisy silences where a time-unit (50 millisecond) is the window shift of sliding spectral segment inputs for prediction).

To evaluate classification on segment files, containing caller and segmentation information, we discretize the continuous segment intervals of calls and noises into 50-millisecond discrete units for predicted and annotated segment files to get two discrete label sequences of the same size.

We evaluate the hypothesized and reference label sequences (comprising 8 call types and the noise type that indicates silence between calls) by counting correctly and incorrectly classified labels. We calculate the accuracy by

$$noise\_acc = \frac{c_{noise}}{n_{noise}}, \tag{1}$$

$$call\_acc = \frac{c_{call}}{n_{call}}, \tag{2}$$

$$total\_acc = \frac{c_{all}}{n_{all}}, \tag{3}$$

where c_{all} is the summation of counts of correct noise labels and call labels, n_{all} is the summation number of call and noise labels (the sequence length); c_{noise} and c_{call} the counts of correct noise and call labels; n_{noise} and n_{call} the total number of noise and call labels. We calculate the precision by

$$precision = \frac{c_{call}}{c_{call} + e_{noise}}, \tag{4}$$

where c_{call} is the number of correct call labels (the same call type for the hypothesis and the reference) and e_{noise} is the number of error noise labels (when predicted as any call but annotated as a noise). We calculate the recall by

$$recall = \frac{c_{call}}{n_{call}}, \tag{5}$$

where n_{call} is the number of call labels in annotated reference. And finally, we calculate the F-score by

$$f = \frac{2 * recall * precision}{recall + precision}. \tag{6}$$

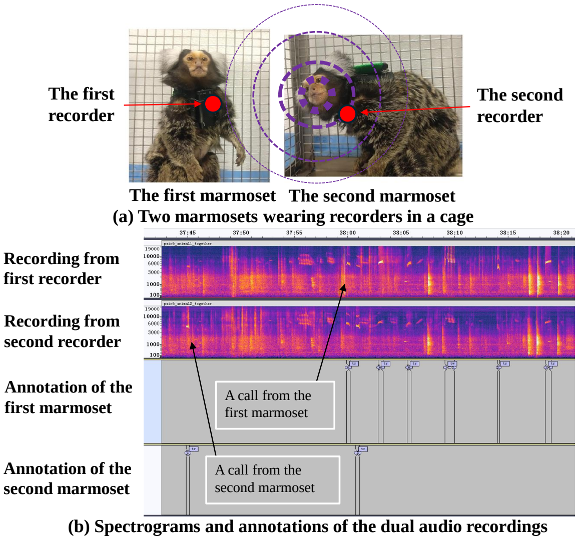
We also evaluate segmentation with the boundary F-score. The boundary F-score is the harmonic mean of boundary precision and recall, where precision is the number of correct intervals over the number of predicted intervals and recall is the number of correct intervals over the number of annotated intervals. A predicted interval is counted as correct only if there exists one annotated interval whose predicted begin and end times match the annotated begin and end times within a given threshold (which we used a tight threshold of 100 milliseconds compared to our model resolution of 50 milliseconds). Each predicted interval is only allowed to match one reference interval.

IV. Dataset and Experimental setup

A. Dataset

We experimented on the dataset [15] designed to study close-range communication between marmosets. We used dual audio recordings from the dataset when two marmosets are together in a cage (excluding recordings when two marmosets are in separate cages). The dataset includes 10 marmoset pairs housed together.

When two marmosets interact each other in a cage while wearing recorders, two simultaneously recorded audios are preserved. The dual-audio recording setup is designed to address the challenging task of call identity annotation. The caller identities were annotated by comparing the spectrograms from the two simultaneous recordings. For example, when the second marmoset makes a call and the first marmoset keeps silent, the recorder worn by the second marmoset should receive a stronger and clearer signal compared to the recorder worn by the first marmoset because the sound wave of the call becomes weaker when it travels further. The close positioning of recorders to the marmosets' mouths is crucial for this approach. Figure 3 demonstrates this process and shows the annotation of call type, call time, and caller identity of a dual audio recording clip.


The first recorder
The second recorder
The first marmoset
The second marmoset
Recording from first recorder
Recording from second recorder
Annotation of the first marmoset
Annotation of the second marmoset
A call from the first marmoset
A call from the second marmoset

(a) Two marmosets wearing recorders in a cage

(b) Spectrograms and annotations of the dual audio recordings

Fig. 3: Upper subfigure (a) Two marmosets wearing recorders in a cage. When the second marmoset vocalizes, the first recorder receives a weaker signal than the second one. Marmoset photos from [15]. Lower subfigure (b) Example clip of an annotated dual audio recording. Calls from other distant cages in the same animal room similarly weak in both audios and are ignored in the annotation [15].

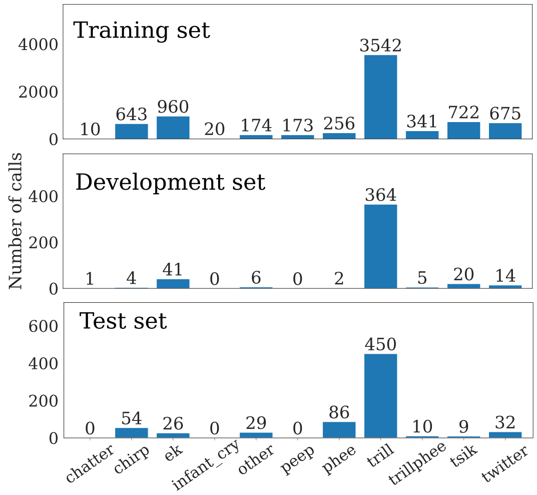


Fig. 4: Number of calls in the training, development, and test sets. The dataset [15] contains recordings of 10 pairs of marmosets, each pair together in a cage. We use the pair1 as the test set, the pair2 as the development set, and the pair3 to pair10 as the training set.

B. Data division

The dataset [15] consists of the dual audio recordings of 10 pairs of marmosets, aged from 1.5 to 10 years old, where the 5 female-male pairs are unrelated and the 5 male-male pairs are siblings. The annotations include 11 call types: trill, twitter, chirp, phee, trillphee, other, ek, tsik, chatter, peep, and infant cry (Figure 4). We divided the dataset with the data of pair1 as the test set, pair2 as the development set, and pair 3 to pair 10 as the training set with duration statistics (Table I) and frequency statistics (Figure 4). All data come from annotated dual audio recordings of marmoset pairs housed together.

TABLE I: Duration of training, development, and test sets of dual-stream annotated marmoset recordings [15] for evaluation of classification, segmentation, and caller identification.

Data division	**Training set**	**Development set**	**Test set**
Number of pairs	8 pairs	1 pair	1 pair
Recording time	16:07:17	1:52:31	2:15:07

TABLE II: Duration of training, development, and test sets for MAE pretraining on marmoset recordings without annotations.

Data division	**Training set**	**Development set**	**Test set**
Number of days	48 days	1 day	1 day
Recording time	639:38:19	17:45:57	18:38:12

We collected longitudinal recordings for 50 days of a marmoset family with parents and one child in a sound-proof box. The recording is single-stream and we recorded more than 10 hours per day. We pretrained the MAE on the whole dataset (Table II) using 48 days for pretraining after roughly removing some empty audios and long silence periods without vocalization. The audios have no labels and we used MAE self-supervised learning on the marmoset linear spectrogram segments without annotations.

C. Experimental setup for systems

We built our systems using the identical 8 call types as [16]. We converted the additional rare call types in the dataset [15] (other, peep, and infant cry) into the noise type. After adding the caller identity information, our target labels become trill, phee, trillphee, twitter, chirp, tsik, ek, chatter (from the first animal), trill2, phee2, trillphee2, twitter2, chirp2, tsik2, ek2, chatter2 (from the second animal), and noise type (indicating intervals between calls for segmentation).

1) Our CNN backbone model: We implemented our backbone CNN model with the same architecture as [16]. It consists of two CNN streams whose outputs are concatenated and linear-projected for the final prediction. Each CNN stream comprises 4 CNN modules. Each module comprises two identical convolutional layers followed by max-pooling. Whenever max-pooling is applied, number of channels in convolutional layers doubles in subsequent module.

2) Proposed Transformer backbone model: We implemented our backbone Transformer model using Vision transformer that divides the original high-resolution 257×256 linear spectral segments into 16×16 patches. These patches are linearly embedded and then passed into a Transformer module with a model dimension of 384. The Transformer module consists of 6 blocks, each with 6-head self-attention and linear modules. The linear modules have a hidden dimension of 1536. The outputs of the two Transformer streams are concatenated and passed through a shared linear layer with a dimension of

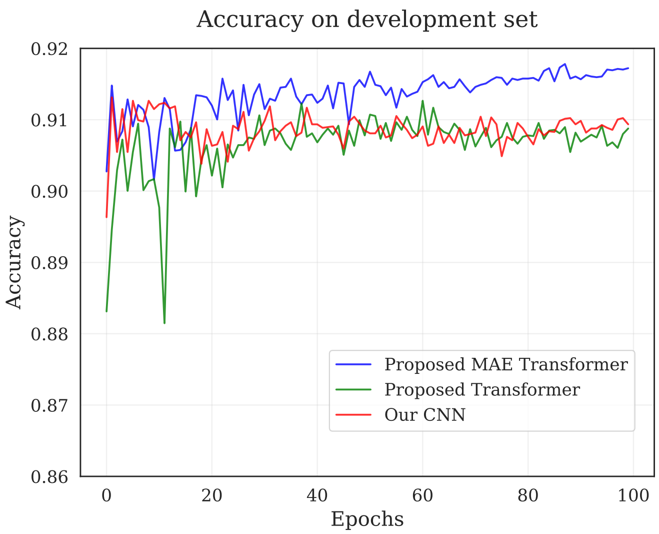


Fig. 5: The accuracy of our CNN, proposed Transformer, and MAE Transformer on the development set during training epochs. The proposed MAE Transformer is the Transformer pretrained using a MAE model on marmoset recordings without labels. We observed that the MAE pretrained Transformer showed better training stability and less overfitting.

1024 for the final prediction. We found that larger Transformer models (e.g., model dimension of 768 or deeper layers) would perform poorly on small amounts of available annotated data.

3) Proposed Transformer backbone model pretrained by MAE: We implemented our backbone Transformer model using an MAE encoder pretrained on the ImageNet training set [19] and 48 days of marmoset spectrogram segments (more than 10 hours per day). We used 12 layers of the pretrained Transformer, with the lower 6 layers frozen used for MAE feature extraction and the upper 6 layers for task-specific fine-tuning. We initialized identical parameters for the Transformers of both streams. We found that using a slightly larger model with model dimension 768 did not degrade performance on limited annotated data due to good pretrained initialization that stabilized training and MAE features that have good generalization power.

The pretrained MAE for marmoset recordings uses 12 layers, 12 attention heads, 768 embeddings with a masking ratio of 75%. The pretraining on the marmoset dataset included 400 training epochs with batch size 256, learning rate 0.00015, 40 warmup epochs, and 0.05 weight decay.

4) Proposed CNN/Transformer system with backbone model: We trained our two-stream system, with CNN or Transformer backbone, to map two 257 × 256 2-dimensional linear spectral segments to the target label. The segments were generated from dual-channel audio using a sliding window of 500ms size with a 150ms shift [16]. The target labeling process was as follows: A call target label was assigned when the middle 150ms part of the segment overlapped with a human-annotated call interval; a noise target label was assigned when the middle 150ms part of the segment did not overlap with any call annotations. These call and noise label targets enable the model to classify and segment long-hour recordings.

TABLE III: Separate classification, caller identification, and segmentation evaluations where segmentation evaluation uses the boundary F-score with 100ms tolerance. We used [16]'s best system as a baseline and evaluated our CNN and proposed Transformer and MAE Transformer systems on the same dataset [15]. MAE_Transformer is the Transformer pretrained using a MAE model on marmoset recordings without labels.

Systems	Call Acc.	Caller Acc.	Boundary F-score
Baseline	0.7212	0.7437	0.0180
CNN	0.7575	0.7866	**0.0182**
Transformer	0.7576	0.7867	0.0152
MAE_Transformer	**0.7811**	**0.8119**	0.0166

To handle long inactive silent intervals in long-hour recordings when marmosets do not vocalize, we implemented two strategies. First, we randomly discarded 4/5 of noise segments to improve training efficiency and balance the dataset. Second, we applied data augmentation by randomly roll-shifting each spectral segment 1-5 pixels vertically and horizontally during model training. These approaches addressed the uneven distribution of vocalizations while enhancing the system's ability to accurately classify marmoset calls.

To use the system to monitor marmoset behavior for long-hour recordings, we applied a streaming technique to enhance model prediction speed. This process involves two main steps: First, we split the test audios into non-overlapping 2500ms segments. Second, we created 50 sub-segments from each 2500ms segment, using a window size of 500ms and a window shift of 50ms: the first 41 sub-segments are complete within the current segment, while the last 9 concatenate parts from the next 2500ms segment. During prediction, after feeding a batch into the model, we obtain predicted 50ms intervals corresponding to 50 subsegments in the batch. This approach, inspired by [16], allows for efficient processing of long-duration audio recordings. We also implemented spectral feature extraction with Pytorch that uses GPU for better efficiency.

We implemented our Transformer system using a Vision Transformer model. Our two-stream Transformer system maintains the same overall architecture as our CNN system but replaces the CNN backbone with a Vision Transformer model. Following [16], our system uses Adam optimizer with a learning rate of 0.0003, decaying by a factor of 0.97 each epoch.

V. Result and Discussion

We applied our system directly to each raw long-hour audio recording (approximately 2 to 3 hours) without extra processing. We compare our PyTorch-implemented two-stream CNN and transformer system with the best sytem of [16] (open-sourced) on the same dataset [15] of annotated dual audio recordings with the data division shown in the Figure 4.

The Figure 5 shows real-time accuracy over 100 epochs on the development set during training of the CNN, Transformer,

TABLE IV: Results of our CNN and proposed Transformer and MAE Transformer systems and [16]'s best system as a baseline on the same dataset [15]. The proposed MAE Transformer is the Transformer pretrained using a MAE model on marmoset recordings without labels.

Systems	**F-score**	**Recall**	**Prec.**
Baseline	0.7686	0.7212	0.8227
Our CNN system	0.7901	0.7575	0.8257
Proposed Transformer	0.7940	0.7576	**0.8341**
Proposed MAE Transformer	**0.7998**	**0.7811**	0.8195

Systems	**Total Acc.**	**Noise Acc.**	**Call Acc.**
Baseline	0.9899	0.9963	0.7212
Our CNN system	0.9907	0.9962	0.7575
Proposed Transformer	0.9909	**0.9964**	0.7576
Proposed MAE Transformer	**0.9909**	0.9959	**0.7811**

and MAE pretrained Transformer where MAE is pretrained with marmoset recordings without annotation. We observed that: 1) For CNN, the small model quickly converges on the limited training data within 25 epochs and then overfits without recovery. 2) For the Transformer model, overfitting occurs much later around 60 epochs, but its starting accuracy is low and shows a steep drop in accuracy due to unstable training. 3) For the MAE pretrained Transformer (6 frozen layers for MAE features and 6 layers for fine-tuning for MAE Transformer vs. 6-layer Transformer), the starting accuracy after the first epoch is even better than CNN. We did not observe any overfitting during fine-tuning, with accuracy continuing to increase until the end of 100 epochs. The MAE pretrained Transformer outperformed both the Transformer and CNN by a large margin.

Our proposed Transformer systems surpass the CNN system and baseline system [16] across all F-score and total accuracy (Table IV). Our proposed Transformer systems also achieve the best performance in separate evaluations of classification, and caller identification (Table III).

The Transformer performs best using global contextual modeling to better capture overall call patterns. However, the Vision Transformer patchizes input spectrograms, resulting in lower model resolution than the CNN model. The transformer prediction intervals are usually wider than annotated truth, penalized by strict segmentation evaluation such as 100ms tolerance boundary F-score (Table III). In future work, we will improve Transformer system for better resolution modeling.

Our systems can be applied to raw, long-hour audio recordings to segment and classify calls, as well as identify callers. This capability provides a valuable tool for recording marmoset interactions, which can support future studies on social behavior, development, and abnormalities in marmoset vocalizations. Such research could offer insights into communication, evolution, and dysfunction of the vocal language of marmoset. The research also helps facilitate comparisons between marmoset and human infant vocal development in family environments. Additionally, the Transformer update of our system offers potential for developing a unified multimodal model that integrates both video and audio inputs.

VI. Conclusion

We proposed self-supervised pretraining that enables robust marmoset call segmentation, classification, and caller identification with two-stream Transformer systems. The Transformer systems outperform previous CNN approaches. Our Transformer pretrained by MAE on hundreds of hours of marmoset recordings without annotations shows the best performance with more stable training and less overfitting compared to CNN and Transformer without pretraining. Our system efficiently processes long-hour or full-day marmoset recordings for vocal communication between marmosets in spontaneous interactions, advancing research on language evolution, development, and dysfunction.

References

[1] J. H. Kaas, *Evolution of nervous systems*. Academic Press, 2016.

[2] H.-C. Chen, G. Kaplan, and L. Rogers, "Contact calls of common marmosets (callithrix jacchus): influence of age of caller on antiphonal calling and other vocal responses," *American Journal of Primatology: Official Journal of the American Society of Primatologists*, vol. 71, no. 2, pp. 165–170, 2009.

[3] 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.

[4] M. Wilson and T. P. Wilson, "An oscillator model of the timing of turn-taking," *Psychonomic bulletin & review*, vol. 12, pp. 957–968, 2005.

[5] J. M. Burkart, S. B. Hrdy, and C. P. Van Schaik, "Cooperative breeding and human cognitive evolution," *Evolutionary Anthropology: Issues, News, and Reviews: Issues, News, and Reviews*, vol. 18, no. 5, pp. 175–186, 2009.

[6] J. M. Burkart and C. P. Van Schaik, "Cognitive consequences of cooperative breeding in primates?" *Animal cognition*, vol. 13, pp. 1–19, 2010.

[7] M. Uesaka, H. Kawauchi, K. Yamaoka, Y. Wakabayashi, Y. Kinoshita, N. Ono, J. Noguchi, S. Watanabe, N. Ichinohe, S. Benner, and H. Yamasue, "Classification of marmoset calls by machine learning and its application to analysis of their vocal development," *Technical report, The 2023 Spring meeting of the Acoustical Society of Japan*, 2023.

[8] N. Kishi, K. Sato, E. Sasaki, and H. Okano, "Common marmoset as a new model animal for neuroscience research and genome editing technology," *Development, growth & differentiation*, vol. 56, no. 1, pp. 53–62, 2014.

[9] D. A. Liao, Y. S. Zhang, L. X. Cai, and A. A. Ghazanfar, "Internal states and extrinsic factors both determine monkey vocal production," *Proceedings of the National Academy of Sciences*, vol. 115, no. 15, pp. 3978–3983, 2018.

[10] D. Takahashi, A. Fenley, Y. Teramoto, D. Narayanan, J. Borjon, P. Holmes, and A. Ghazanfar, "The developmental dynamics of marmoset monkey vocal production." *Science*, vol. 349, no. 6249, pp. 734–738, 2015.

[11] H. K. Turesson, S. Ribeiro, D. R. Pereira, J. P. Papa, and V. H. C. de Albuquerque, "Machine learning algorithms for automatic classification of marmoset vocalizations," *PLOS ONE*, vol. 11, no. 9, p. e0163041, Sep. 2016.

[12] A. Wisler, L. J. Brattain, R. Landman, and T. F. Quatieri, "A framework for automated marmoset vocalization detection and classification." in *INTERSPEECH*, 2016, pp. 2592–2596.

[13] Y.-J. Zhang, J.-F. 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, no. 1, pp. 478–487, 2018.

[14] E. Sarkar and M. Magimai.-Doss, "Can Self-Supervised Neural Representations Pre-Trained on Human Speech distinguish Animal Callers?" in *INTERSPEECH*, 2023, pp. 1189–1193.

[15] R. Landman, J. Sharma, J. B. Hyman, A. Fanucci-Kiss, O. Meisner, S. Parmar, G. Feng, and R. Desimone, “Close-range vocal interaction in the common marmoset (Callithrix jacchus),” *PLOS ONE*, vol. 15, no. 4, 2020.

[16] T. Oikarinen, K. Srinivasan, O. Meisner, J. B. Hyman, S. Parmar, A. Fanucci-Kiss, R. Desimone, R. Landman, and G. Feng, “Deep convolutional network for animal sound classification and source attribution using dual audio recordings,” *The Journal of the Acoustical Society of America*, vol. 145, no. 2, pp. 654–662, 2019.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NIPS*, 2017, pp. 5998–6008.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *ICLR*, 2021.

[19] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *CVPR*, 2022, pp. 16 000–16 009.