# FEW SHOT BIOACOUSTIC DETECTION BOOSTING WITH FINETUNING STRATEGY USING NEGATIVE-BASED PROTOTYPICAL LEARNING

*Yuna Lee, HaeChun Chung, JaeHoon Jung*

AI2XL Lab.,
Institute of Convergence Technology,
KT Corporation

## ABSTRACT

Sound event detection involves the identification and temporal localization of sound events within audio recordings. Bioacoustic sound event detection specifically targets animal vocalizations, which necessitate substantial time and resources for manual annotation of temporal boundaries. This paper aims to address the challenges associated with bioacoustic sound event detection by proposing a novel prototypical learning framework. Our approach fuses contrastive learning and prototypical learning to use the limited amount of dataset at its utmost. Further, our framework leverages finetuning strategy with a novel loss function to develop a robust framework. Experimental results on a benchmark dataset demonstrate the effectiveness of our proposed method in accurately detecting and localizing bioacoustic sound events, improving the F1 score from 29.59% to 83.08%.

***Index Terms***— Few-shot Learning, Contrastive Learning, finetuning, bioacoustic sound Event Detection

## 1. INTRODUCTION

Sound event detection is the task of recognizing the sound events and their respective temporal start and end times in a recording [1]. In the case of bioacoustic sound event detection, the task focuses on animal vocalizations, which demand time and resources to annotate each time stamp [2]. Few-shot learning (FSL) is a supervised learning method that can achieve high performance on data from completely different domains even with a small amount of data. As all of these tasks encounter data scarcity and the difficulty of building a framework generalized in the acoustic domain, FSL has come into the limelight. In the previous DCASE-T5 challenges, submitted systems achieved great performance by using the transductive inference method [3, 4, 5], improved prototypical learning [6], contrastive learning [7], and multi-class classification learning via splitting the audio segment into frame-level [8]. Nevertheless, proposed methods showed relatively low performance on the evaluation dataset compared to the performance obtained on the validation set. The majority of existing methods adopted prototypical learning to identify positive classes from negative classes. Prototypical learning itself demonstrated high performance, there were two limitations to taking the performance to another level. Firstly, the capability of high-level feature learning was challenging since the model was trained on classifying binary classes, which are positive and negative. Second, the loss function of current prototypical learning [9] focuses on pulling positive classes, which we refer to as "positive-based prototypical loss function (PPL)". It may be promising on the training dataset, which contains a sufficient amount of positive class data, but it can lead to overfitting when the amount of negative class data is much greater than that of positive class. If the model is trained in the standard prototypical learning manner, the embedding features of negative classes are highly likely to be dispersed, while those of positive classes are well-clustered in the embedding space. As the class imbalance problem is prevalent in the bioacoustic domain, we propose a fine-tuning strategy with a negative-based prototypical loss function (NPL) to ameliorate this issue. The proposed method suggests additional training on negative class data to enhance the ability to aggregate negative classes in the embedding space. By applying the proposed strategy, the pretrained model can attain the superior capability to discriminate between positive and negative classes. Through this strategy, the pretrained model can achieve a higher F-measure on the validation dataset.

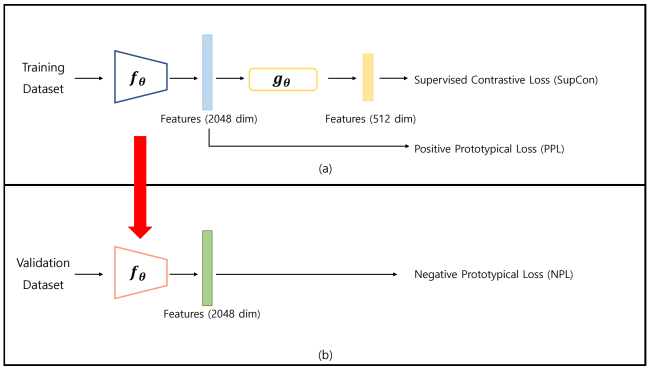

Figure 1: Overview of the proposed framework. The framework consists of a pretraining stage and finetuning stage. The pretraining stage is described in (a). The encoder $f_\theta$ is trained on the training dataset through supervised contrastive loss (SupCon) and PPL functions. Also, the finetuning stage can be seen in (b). Pretrained encoder $f_\theta$ is finetuned on the validation dataset. We exploit NPL function throughout finetuning process.

## 2. METHODS

### 2.1. Outline

Our overall framework can be shown in Figure 1. We utilize our method in $N$-way $K$-shot task. Prior to previous systems [3, 4, 5, 6, 7, 8], we denote the positive segment as the target sound event and the negative segment as the audio segments that do not contain the target sound event in each audio file. Given the fact that training dataset contains 45 classes and task 5 is regarded as 5-shot learning problem, we set $N = 45$ and $K = 5$. As each audio file in the validation dataset should be considered independently, we define negative segments from a single audio file as solitary negative classes instead of grouping negative segments into a single 'unknown' class. Simply put, each audio file contains a single positive class and a single negative class. Also, our system has 45 negative classes along with 45 positive classes. This enables encoder network $f_\theta(\cdot)$ to cluster positive segments more densely, maximizing the gap between positive segments and negative segments.

### 2.2. Pretraining Stage

In the pretraining stage, we train the encoder network $f_\theta(\cdot)$. We select each $2 \times K$ positive segments and negative segments from the dataset and set $K$ segments as support segments and the other as query segments. We denote the positive support set of class $i$ as $S_i^p$ and the query set as $Q_i^p$, and the negative support set and the query set of class $i$ can be expressed as $S_i^n$, $Q_i^n$ where $|S| = |Q| = K$. The prototype of each set defined in class $i$, which is represented by the mean embedding vectors, is defined as the equation below.

$$s_i^* = \frac{1}{|S_i^*|}\sum_{(x_i,y_i)\in S_i^*} f_\theta(x_i), q_i^* = \frac{1}{|Q_i^*|}\sum_{(x_i,y_i)\in Q_i^*} f_\theta(x_i) \quad (1)$$

where $(x_i, y_i)$ are the segment and its label of the class $i$ in each set. Equation 2 describes $PP_j^i$, which is the euclidean distance between positive embedding vectors of $Q_i^p$ and positive support prototype of class $j$, $s_j^p$.

$$PP_j^i = \left(\sqrt{\sum_{x\in Q_i^p}(f_\theta(x) - s_j^p)^2}\right) \quad (2)$$

In the same way, we denote $PN_j^i$, which is the euclidean distance between embedding vectors of $Q_i^p$ and negative support prototype $s_j^n$.

$$PN_j^i = \left(\sqrt{\sum_{x\in Q_i^p}(f_\theta(x) - s_j^n)^2}\right) \quad (3)$$

We can formulate positive-based loss for class $i$ as the equation below.

$$ppl_i = -log\left(\frac{exp\left(-PP_i^i\right)}{\sum_{j=1}^N \left(exp\left(-PP_j^i\right) + exp\left(-PN_j^i\right)\right)}\right) \quad (4)$$

Using equation 4, PPL is described as the equation 5.

$$PPL = \frac{1}{N}\sum_{i=1}^N ppl_i \quad (5)$$

We pretrain $f_\theta(\cdot)$ with PPL function and supervised contrastive (SupCon) loss function [10] to enhance the feature representation capacity of $f_\theta(\cdot)$. We build a 2-layer projection layer $g_\theta(\cdot)$ to create embedding vectors for each audio segment in the following step. Thus, our total loss function for pretraining step is $\mathcal{L}_{train} = \mathcal{L}_{PPL} + \mathcal{L}_{SupCon}$. We adopt 3-layer ResNet [11] network from previous years' method [3] as $f_\theta(\cdot)$. We set the output embedding dimension to 2048 for $\mathcal{L}_{PPL}$, and downsize the dimension to 512 for $\mathcal{L}_{SupCon}$. Through the pretraining stage, the encoder network $f_\theta(\cdot)$ can attain the ability to embed positive classes well in the embedding space.

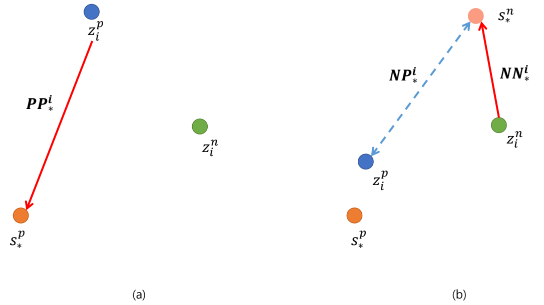

Figure 2: Let $z_i^p = f_\theta(x_i^p)$ be the positive embedding vector in the query set of class $i$, and $z_i^n = f_\theta(x_i^n)$ as the negative embedding vector in the following set. (a) depicts PPL function, which seeks to minimize $PP_*^i$. (b) describes the NPL function, minimize $NN_*$ while maximize $NP_*^i$. Given that the encoder network already possesses the capability to cluster positive classes, we utilize NPL during the fine-tuning stage to increase the distance between $s_*^n$ and $s_*^p$. The red line infers pull force, and the blue dotted line refers to push force.

### 2.3. Finetuning Stage

After the pretraining stage, $f_\theta(\cdot)$ is capable of detecting positive segments from negative segments. Nonetheless, the dataset is comprised of a large number of negative segments and a scarce amount of positive segments in the bioacoustic domain. This fact may not guarantee the good performance of $f_\theta(\cdot)$ on the general bioacoustic domain. In order to resolve data scarcity and performance maintenance issues, we figured that a sole training stage was not enough. Based on the unique characteristic of the bioacoustic dataset, we finetune $f_\theta(\cdot)$ to aim at negative-based feature learning, which is the opposite of the aforementioned stage. We display a comparison of PPL and NPL in Figure 2. Further, we propose a further developed Distance-based NPL function by incorporating the Furthest Point Sampling (FPS) algorithm [12] into the NPL function.

**Negative-based Prototypical Loss** We add an additional definition of distances between embedding vectors of $Q_*^n$ and support prototypes. Unlike PPL, NPL minimizes the distance between negative embedding vectors and $s^n$ while maximizing the distance between the positive embedding vectors. We redesign the positive-based loss $ppl_i$ as the equation 6.

$$pnl_i = -log\left(\frac{exp\left(PN_i^i\right)}{\sum_{j=1}^N \left(exp\left(PP_j^i\right) + exp\left(PN_j^i\right)\right)}\right) \quad (6)$$

Following the equations 2 and 3, we define $NP$ and $NN$ as euclidean distance of negative query embedding vectors between the positive support prototype and negative support prototype. Equation 7 and 8 describes $NP$ and $NN$ specifically.

$$NP_j^i = \left(\sqrt{\sum_{x\in Q_i^n}(f_\theta(x) - s_j^p)^2}\right) \quad (7)$$

$$NN_j^i = \left(\sqrt{\sum_{x\in Q_i^n}(f_\theta(x) - s_j^n)^2}\right) \quad (8)$$

And we add new negative-based loss $nnl_i$ to minimize the gap between negative embedding vectors and $s^n$. The following distance function is described below.

$$nnl_i = -log\left(\frac{exp\left(-NN_i^i\right)}{\sum_{j=1}^{N}\left(exp\left(-NP_j^i\right) + exp\left(-NN_j^i\right)\right)}\right) \quad (9)$$

With equations 6 and 9, NPL function can be summarized as equation 10.

$$NPL = \frac{1}{N}\sum_{i=1}^{N}(pnl_i + nnl_i) \quad (10)$$

By finetuning $f_\theta(\cdot)$ with $\mathcal{L}_{NPL}$, $f_\theta(\cdot)$ learns the ability to cluster negative embedding vectors and negative prototype more densely, giving the effect of separating positive segments and negative segments.

**Distance-based Negative-based Prototypical Loss** While NPL loss function randomly picks $K$ support features and $K$ query features from $2 \times K$ arbitrarily chosen features, we extend NPL loss function by adopting the idea of the Furthest Point Sampling (FPS) algorithm. FPS algorithm is a classic method used in 3D point clouds [12]. Since we aim to clump negative embedding vectors and negative prototype, we believe the distance-based selection of query and support features can maximize the efficacy of NPL loss function. All distances between $2 \times K$ randomly extracted positive features and $2 \times K$ negative features are calculated. The positive and negative features with the shortest distance are selected as a pair of reference features. Nearest-neighbor sampling method [13] is attempted based on the selected positive reference feature and negative reference feature. Thus, we set negative features placed close to the positive features as a negative support set, and positive features closely located to the negative features as a positive query set. Then, we optimize the loss function to maximize the distance between the negative prototype and positive query set so that we can ultimately maximize $PN$. We conduct the furthest sampling based on the prior negative reference feature in negative features. Through this process, negative features located on the outskirts will be selected from negative features, and non-selected features will be located on the inner side among negative features. We set the selected features to a negative query set and the unselected features to a negative support set. The negative prototype created from negative support set is used to minimize the distance between negative query set, eventually minimizing $NN$. In this way, we can boost the initial goal of NPL by optimizing the maximization of positive-negative distance and minimization of negative-to-negative distance at the same time. The following procedures are illustrated in Figure 3. For post-processing and inference, we applied methods proposed in the DCASE 2022 challenge [3].

## 3. EXPERIMENT

### 3.1. Experimental setup

We conducted the experiments for two purposes. First, we prove that our novel framework is more applicable in the few-shot learning domain than baseline methods. In the previous DCASE challenges, transductive inference (TI) method adapted from [14] played a crucial role in challenge [15, 16, 17, 18]. Here, we apply part of the TI method as a variant to our scheme. Thus, we compare variants with our method to analyze the impact of our novel finetuning strategies as an ablation study. Second, we intend to prove the efficacy of our proposed method by comparing the results of grafting the finetuning strategy. We set contrastive learning and few-shot learning as our basic framework. In all experiments, the learning rate was set to 0.001 and the input length was fixed at 0.2 seconds. To prevent overfitting on any dataset, we implemented early stopping. We did not use any augmentation or additional acoustic features. We adopted the official evaluation metric[1] as our evaluation metric. Since the full annotation of the evaluation set was not released in public, we considered the validation set of the DCASE 2023 task 5 dataset as the evaluation set.

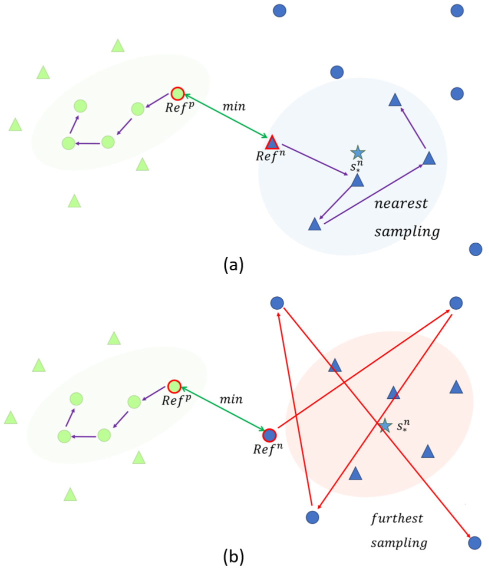

Figure 3: We denote each positive and negative reference feature as $Ref^p$ and $Ref^n$. The triangle, circle, and star-shaped figure each represent the feature vectors of the support set, the query set, and the prototype respectively. (a) shows the process of maximizing PN based on $Ref^*$ through nearest-neighbor sampling. (b) is the process of minimizing the NN via the furthest point sampling.

| | | Precision (%) | Recall (%) | F-measure (%) |
|---|---|---|---|---|
| Template Matching | | 2.42 | 18.32 | 4.28 |
| Prototypical Network | | 36.34 | 24.96 | 29.59 |
| DCASE2022 Winning Team [8] | | 77.50 | 71.50 | 74.40 |
| **Ours** | Pretraining | 74.27 | 56.70 | 64.31 |
| | Finetuning | 89.93 | 77.20 | **83.08** |

Table 1: The precision, recall, and F-measure of the validation set.

## 4. RESULTS

### 4.1. Performance Comparison

We compare our methods with baseline schemes and the winning team of DCASE 2022 [8]. we describe our basic framework as the "backbone" for convenience. Pretraining denotes the performance of the encoder $f_\theta(\cdot)$ after the pretraining stage, and Finetuning denotes the performance after the finetuning stage. As can

[1] https://github.com/c4dm/dcase-few-shot-bioacoustic

| System | PB | ME | HB | Overall | | |
|---|---|---|---|---|---|---|
| | F-measure (%) | | | Pre (%) | Rec (%) | F-measure (%) |
| Backbone | 45.27 | 74.58 | 80.66 | 66.39 | 59.28 | 62.64 |
| w. TI method | 47.18 | 85.71 | 72.50 | 74.27 | 56.70 | 64.31 |
| w. Distance-based NPL finetuning | 63.04 | 95.41 | 95.60 | 90.17 | 74.38 | 81.52 |
| w. TI method & Distance-based NPL finetuning | 63.45 | 99.05 | 97.53 | 89.93 | 77.20 | **83.08** |

Table 2: The precision, recall, and f-measure of each subset in the validation set.

be seen in Table 1, our proposed method outnumbers both baseline and 2022 challenge-winning team by a large margin. We also evaluated our encoder network $f_\theta(\cdot)$ after each stage to confirm the impact of the distance-based NPL function. The performance disparity between the two stages clearly verifies distance-based NPL function actually have a meaningful impact on developing the capacity to detect positive sound event even in the highly imbalanced dataset, increasing the performance up to 18.77%. In Table 2, we compare our basic scheme and its variants. We select the condition where the distance-based NPL finetuning strategy and TI method are additionally applied to our backbone for comparison. We select systems with different conditions as mentioned in section 3.1. Our system showed relatively low performance on the PB dataset relative to other datasets in general. We assume this phenomenon is due to the drastic ratio between the positive segment and the negative segment as it contains a relatively short duration of the positive segment. Since the features extracted from positive segments are limited, the encoder network $f_\theta(\cdot)$ finds it more difficult to detect positive segments. This phenomenon was consistently observed in the performance of the DCASE2023 evaluation dataset [19]. All of our submitted systems showed relatively low performance on the CT dataset, in which the majority of positive segments are less than 0.2 seconds, which is the minimum input length of our method.

### 4.2. Ablation Study

In the ablation study, we compare our baseline scheme and the combination of two different novel finetuning strategies. We compared the case where only the basic training stage was performed for each baseline and the case where original NPL finetuning and distance-based NPL finetuning was applied.

| System | F-measure |
|---|---|
| Backbone | 62.64 |
| **w.** NPL finetuning | 79.79 |
| **w.** Distance-based NPL finetuning | **81.52** |

Table 3: Ablation study of the proposed method.

Table 3 states that finetuning strategy with the NPL function and the distance-based NPL function shows a noticeable numerical difference. We presume the following difference is based on the prototype selection. While typical NPL selects support features and query features randomly, distance-based NPL is based on euclidean distance, which is more definite. This induces the network to finetune in a way that estimates the position of the positive prototype and escalates $PN$, increasing performance more intuitively. The effect of distance-based NPL finetuning is visualized with t-SNE [20] in Figure 4. As shown in figure 4, t-SNE of the same class tend to cluster more densely after distance-based NPL is exploited. The fact that the distance-based NPL performed better than the conventional NPL was also confirmed in the performance of the evaluation set. It was confirmed that the systems finetuned with distance-based NPL performed better than the systems finetuned with typical NPL. The performance gap was more prominent in the case where the positive class and the negative class were very similar, such as MGE dataset in the DCASE 2023 evaluation set [19].

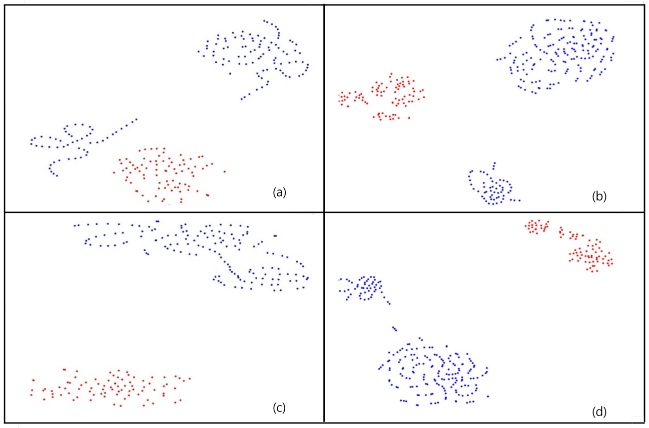

Figure 4: (a) and (c) are extracted from the same audio, and (b) and (d) are extracted from the same audio file. (a), (b) are visualizations extracted after pretraining stage. and (c), and (d) are extracted after finetuning stage. Red dot represents embedding vectors extracted from positive segments, and blue dot represents vectors extracted from negative segments.

## 5. CONCLUSION

In this paper, we presented a novel framework for few-shot bioacoustic event detection. Our method combines the contrastive learning method and prototypical learning and uses the novel finetuning strategy of using a modified prototypical loss function. The proposed pretraining process enables embedding positive class data on the embedding space, NPL finetuning strategy enables pretrained network to detect sound events in the environment where positive sound events were unseen in the training stage or fine-tuning stage. Experiments showed that the proposed framework can robustly separate positive and negative segments in highly imbalanced datasets. Further, the fact that all of the submitted systems achieve high F-measure scores on two new subsets proves its ability to generalize to new classes [19].

## 6. REFERENCES

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[2] http://dcase.community/challenge2023/.

[3] H. Liu, X. Liu, X. Mei, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey system for dcase 2022 task 5 : Few-shot bioacoustic event detection with segment-level metric learning technical report," DCASE2022 Challenge, Tech. Rep., June 2022.

[4] Y. Tan, L. Xu, C. Zhu, S. Li, H. Ai, and X. Shao, "A new transductive framework for few-shot bioacoustic event detection task," June 2022.

[5] Q. Huang, Y. Li, W. Cao, and H. Chen, "Few-shot bio-acoustic event detection based on transductive learning and adapted central difference convolution," June 2022.

[6] D. Yang, Y. Zou, F. Cui, and Y. Wang, "Improved prototypical network with data augmentation," June 2022.

[7] B. Zgorzynski and M. Matuszewski, "Siamese network for few-shot bioacoustic event detection," June 2022.

[8] J. Tang, X. Zhang, T. Gao, D. Liu, J. P. Xin Fang and, Q. Wang, J. Du, K. Xu, and Q. Pan, "Few-shot embedding learning and event filtering for bioacoustic event detection," June 2022.

[9] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[10] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

[13] B. W. Silverman and M. C. Jones, "E. fix and jl hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951)," *International Statistical Review/Revue Internationale de Statistique*, pp. 233–238, 1989.

[14] M. Boudiaf, I. Ziko, J. Rony, J. Dolz, P. Piantanida, and I. Ben Ayed, "Information maximization for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2445–2457, 2020.

[15] D. Yang, H. Wang, Y. Zou, Z. Ye, and W. Wang, "A mutual learning framework for few-shot sound event detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 811–815.

[16] M. Boudiaf, H. Kervadec, Z. I. Masud, P. Piantanida, I. Ben Ayed, and J. Dolz, "Few-shot segmentation without meta-learning: A good transductive inference is all you need?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 979–13 988.

[17] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," *arXiv preprint arXiv:1805.10002*, 2018.

[18] D. Yang, H. Wang, Z. Ye, and Y. Zou, "Few-shot bioacoustic event detection= a good transductive inference is all you need," DCASE2021 Challenge, Tech. Rep, Tech. Rep., 2021.

[19] I. Nolasco, S. Singh, E. Vidana-Villa, E. Grout, J. Morford, M. Emmerson, F. Jensens, H. Whitehead, I. Kiskin, A. Strandburg-Peshkin, *et al.*, "Few-shot bioacoustic event detection at the dcase 2022 challenge," *arXiv preprint arXiv:2207.07911*, 2022.

[20] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.