# Prototypical Network Regularization using KL Divergence

Qi Wen Gan, Jimmy ZhiCong Lin, Futang Wang
Department of Computer Science and Technology
Tsinghua University
{yanqm21, zc-lin21, wft21} @mails.tsinghua.edu.cn

## Abstract

*In this report, we describe our proposed new loss function for DCASE2021 Task5: few-shot bioacoustic event detection. We come out with a novel proposal of Kullback-Leibler divergence (KL divergence) loss regularization method on the top of deep learning prototypical network. During episodic training, the new loss updates the feature extractor to diversify the selection of components for a given class, thus, the model will output a more evenly distributed prediction to each unseen class. On the official provided validation set, we demonstrate that the proposed method achieves the overall F-measure score of 45.61 %, with about 5 % improvement over the baseline on the validation set.*

## 1. Introduction

The few-shot bioacoustics event detection task is a part of the 2021 Detection and Classification of Acoustic Scenes and Events (DCASE 2021) competition. The task focuses on classifying animals (birds and mammals) through vocalizations. The main objective of this task is to find a reliable algorithm that is capable of dealing with data sparsity, class imbalance, and noisy/busy environments. Ultimately, this task will help discover the potential of few-shot learning as it can be an extremely good fit for the needs of users in bioacoustics and also the wider domain of sound event detection (SED). Recently, prototypical networks also successfully applied to few-shot speech recognition [1]

## 2. Background and Related Work

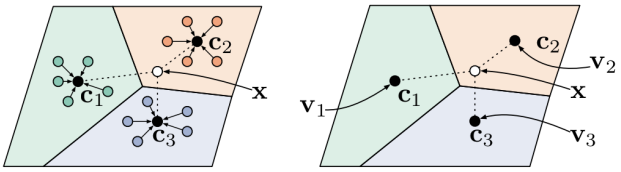
In the following section, we explain the task of few shot learning and its importance in the progression of deep learning. Additionally, we will discuss the baseline model from DCASE 2021 Task 5 which we compare with to evaluate the effectiveness of our model.

### 2.1. Few shot learning

Few-shot learning is a highly promising paradigm for sound event detection. It is also an extremely good fit to the needs of users in bioacoustics, in which increasingly large acoustic datasets commonly need to be labeled for events of an identified category. The main objective of this task is to find a reliable algorithm that is capable of dealing with data sparsity, class imbalance, and noisy/busy environments. Research in few-shot learning has received growing interest as supervised data is scarce and can lead traditional models to make unreliable generalizations. Additionally, it is costly in terms of time and resources for labeling large datasets, hence few-shot models can reduce such costs whilst maintaining classification accuracy. Model-Agnostic Meta-Learning (MAML) offers a solution to few-shot learning as its goal is to train a model on a variety of learning tasks, such that it can solve new learning tasks using only a small number of training samples [2]. Ultimately, we believe few-shot learning is extremely useful in deep learning and have decided to study it through this project.

### 2.2. Prototypical Network

The baseline model for DCASE 2021, utilises a prototypical network structure. Prototypical networks are based on the idea that there exists an embedding in which points cluster around a single prototype representation for each class [8]. In order to do this, we learn a non-linear mapping of the input into an embedding space using a neural network and take a class's prototype to be the mean of its support set in the embedding space. Classification is then performed for an embedded query point by simply finding the nearest class prototype.
The prototypical networks are coupled with episodic training which utilizes sampled mini-batches called episodes during training, where each episode is designed to mimic the few-shot task by subsampling classes as well as data points. The use of episodes makes the training problem more faithful to the test environment and thereby improves generalization. For the baseline model, the positive annotations in the training data are of unequal duration, hence




(a) Few-shot (b) Zero-shot

Figure 1. Left diagram shows the classification for few shot learning and the right shows the classification for zero-shot learning. In both cases $c_k$ represents prototypes classes and the embedded query points $X$ are classified through a soft-max over distances to prototypes.

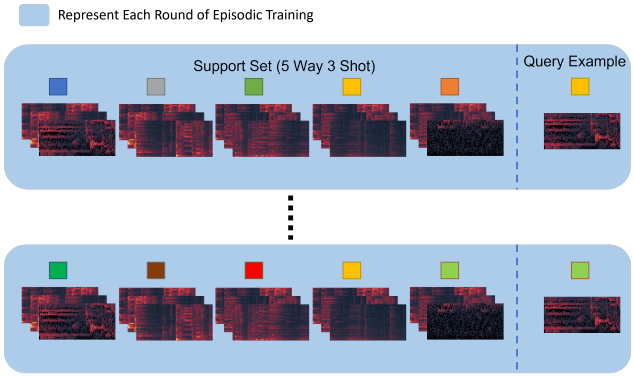



Figure 2. For each episodic training, we minimize the distant between the $Y_{pred}$ distribution from and the $Y_{Target}$ distribution. Hopefully, during the evaluation or inference stage, the model will output a more evenly distributed prediction to each unseen class.

the model extracted equal length patches from the annotated segments, where each patch inherits the label of its corresponding annotation [1].

## 3. Methods

In this section, we first explain the few shot settings for the sound event detection task. We dissect the Then, we will discuss the overall idea of our new proposed loss function formulation and explain the intuition behind guiding the prototypical network learning.

### 3.1. Few-shot settings

The goal of the few shot settings in sound event detection tasks is to learn a classifier given a few labeled unseen classes. In few shot settings, the problem is always described as N-way K-shot where N-way represent the number of class, and K-Shot-classification problem refer to K-labeled images for each class. The K-labeled image is usually a small amount, less than ten samples per class. During training, given the training set with total number of classes and total number of samples $C_{train}, X_i \in X_{train}$. We samples support set $S = (s_i, ..., s_N)|s_i \in \{x_1, ..., x_{k\times c}, x_q\}$ consists of small support examples and a query example $x_q$. where x is the input feature and N is the number of support sets. The support examples are randomly sampled k examples from each of the N classes in the training set D, and the query sample is randomly chosen from the remaining examples of c classes. It is important to know that the query sample $x_q$ will always come from one of the sampled support classes $y_q \in S_N$. Please refer to figure 2 for details. This method is also called "episodic training" and the operation will be reused during the inference time.

### 3.2. Proposed Loss Function

In the prototypical network paper, Snell *et al*. [8] experimented with substituting the Euclidean distance with cosine similarity distance that is widely used in representation learning nowadays [6]. However, the results show that by using the cosine similarity distant function, the results are not as good as the Euclidien distant function. Therefore, the author further hypothesized that any kind of Bregman divergences distant function will work well in any similar few shot learning task. Inspired by this quote, we found that KL Divergence [5], a member of Bergman divergence is feasible and works well during our experiment settings. Different from the state-of-the-art algorithms [2, 9, 12]for few-shot learning, our loss function is modified based on the original Euclidean Loss. However, it is natural to ask where do we adding this KL divergence Loss function?

Intuitively, the additional KL Divergence Loss, $KL_{loss}$ is to minimize the 'distance' of two distribution $Y_{Pred}$ and $Y_{Target}$ during single episodic training, where $Y_{Pred}$ is the prediction of the network and $Y_{Target}$ is the ground truth labels distribution for the query set.

In our experiment, model parameters are updated using two losses: the first "$Euclid_{loss}$" loss which updates both the feature extractor such that query feature vectors are assigned to their nearest class feature component; and the "$KL_{loss}$" loss, which updates the feature extractor to diversify the selection of components for a given class. During training, this is done by utilizing the $Y_{Target}$ distribution that is set to be a constant throughout the training stage. For each episodic training, we always draw 50 query samples that come from 10 distinct classes and 5 samples for each distinct class. After applying the inverse distance of each query sample to the nearest prototypical embedding, we can get the log probability of classifier prediction of each 50 queries sample, $Y^i_{Pred}$. Based on empirical observation, the "distant" of two distributions $Y_{Pred}$ and $Y_{Target}$ are generally small. Thus, we can simply add the $KL_{loss}$ loss on the top of the original $Euclid_{loss}$.

We further formulate the Loss function of the Learning Task with the equation below:


Figure 3. For each episodic training, we minimize the distant between the $Y_{pred}$ distribution from and the $Y_{Target}$ distribution. Hopefully, during the evaluation or inference stage, the model will output a more evenly distributed prediction to each unseen class.

$$\mathcal{L} = KL_{loss} + Euclid_{loss} \tag{1}$$

$$Euclid_{Loss} = \sum_{i=1}^{n} (q_i - p_i)^2 \tag{2}$$

$$KL_{Loss}(\hat{y}||y) = \hat{y}_c \log \frac{\hat{y}_c}{y_c} \tag{3}$$

where $q_i$ denotes query samples, and $p_i$ prototypes, $\hat{y}$ and $y$ represent the predicted distribution $Y_{Pred}$ and ground truth distribution$Y_{Target}$ after applied the Log Softmax operation.

# 4. Experiments

In this part, we will first introduce the dataset we used in our experiments. Then we will introduce the quantitative metrics to evaluate the system and lastly report the experiment results of our proposed method.

## 4.1. Dataset

In this challenge, the development set provided by the official organizers [4] is pre-split into training and validation sets. The training set consists of four different sub-folders (BV, HV, JD, MT), each for one source class. Along with the audio files multi-class annotations are provided for each. The total duration of the whole training set is 14.3 hours, with 10 hours, 3 hours, 10 minutes, and 1.16 hours for BV, HV, JD, and MT respectively. The total number of classes is 19, in which 11 for BV, 3 for HT, 1 for JD and 4 for MT. In addition, the sampling rate is also very different for different sources, it varies from 6kHz (for HT) to 24kHz (for BV). The validation set comprises two sub-folders (HV, PB). It includes a total 5 hours of data, and covers 4 classes, 2 for HV with 6kHz sampling rate and 2 for PB with 44.1kHz sampling rate. The two classes for each source are actually the target events and the backgrounds.

**Features :**All of our experiments use the same Per-channel energy normalization (PCEN) [11] features as used in the official baseline system [4]. They aim to improve the robustness of mel-frequency spectrogram to channel distortion, by combining dynamic range compression (DRC) and adaptive gain control (AGC) with temporal integration. The PCEN is conducted on mel frequency spectrogram and used as an input feature. Raw audio is scaled to the range $[2^{31}, 2^{31} - 1]$ Before the mel transformation. Detail parameters can be found in [4]

## 4.2. Metrics

For all the experiments, we use the event-based F-measure as the evaluation metric, which is one of the most commonly used metrics for sound event detection. The calculation of F1-Score can be summarized in equation 4, 5, and 6.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

During the evaluation, by using the trained prototypical network, we predict the class of interest with 5-shot setting. We use the first five POS annotations for the class of interest for each file when trying to predict the rest. Therefore, if the system predicted a NEG for POS labeled data, it will be considered as a False Negative. In our new proposed KL divergence loss, the results proved to decrease the False Negative rate in the recall, thus, boosting the overall F1 score of the system. We run our evaluation method provided by DCASE official organizer. Before evaluation, we also run a round of post-processing to get a CSV output from the prototypical network.

## 4.3. Results

All our experiments are performed on the development dataset of DCASE 2021 Task5 Challenge. Two baselines were provided for this task. Here were two baselines systems: Prototypical and Template Matching We have chosen to compete with the results of the baseline prototypical network with the 10 ways 5 shots setting because template matching is not a deep learning approach. We first

re-implement the official baseline using the deep learning approach. We get a similar baseline result that was reported in DCASE 2021 Task5 website [3]. All of the submitted systems are examined on the validation set that the prototypical have not trained on. Results are shown in Table 1. To prove the effectiveness of our new proposed new loss function, We followed one of the other contestant's works [10], which implemented Resnet12 as a backbone network and proved to be a better choice compared to the default 4-layer CNN settings from the original work. [8]

Table 1. Experiments result

| | Network | Precision | Recall | F1-score |
|---|---|---|---|---|
| Baseline | Prototypical | **0.56** | 0.32 | 40.36 |
| Ours | CNN+CLIP$_{Loss}$ | 0.01 | 0.26 | 14.37 |
| | CNN+KL$_{Loss}$ | 0.53 | 0.35 | 42.30 |
| | Resnet12 | 0.46 | 0.40 | 43.11 |
| | Resnet12+KL$_{Loss}$ | 0.53 | **0.40** | **45.61** |

In Table 1, the "CNN" represents the model structure is the 4-layer CNN prototypical network. "ResNet12" represents use 12-layer ResNet [7] replace the 4-layer CNN in prototypical network. The "+clip" represent the substitution of the Euclidien Distant loss function with the Clip loss function as we proposed in our proposal [6]. The $+KL_{Loss}$ represents we add an extra proposed KL loss term on the top of the original Euclidean Distant loss function during each episodic training. Our best results achieved 45.61 percent of F1-score by sacrificing a small amount of precision with a greater amount of recall, hence boosting the F1-score.

### 4.4. Ablation

To further prove our proposed regularization method, we intended to implement our method on Zou *et al*. [12] works, DCASE 2021 Task 5 first-place winner. However, during our experiment based on the author's code, we cannot reproduce their result. Using the given code to train the model, we get an average f1-score evaluation below 40, which is worst than the baseline result. Even when we applied the trained weights given by the author, we can roughly get a 51.34 F1-score, which is 5 percent lower than the result reported on DCASE 2021 website [3]. Due to the reason we are unable to reproduce their result, we are forced to abandon this plan.

## 5. Conclusion

In our experiments, we proposed to add a KL Divergence Loss function on the top of the original proposed Euclidean Distance to enhance the Recall and F1-Score of the SED task. The proposed KL loss function act as a regularizer for smoothing the distribution of the network predicted query output with the target query output during each episodic training. Our experiment results show that we can achieve a 5.25 percentage of improvement when we implemented Resnet12 as our backbone network and use the proposed new KL Loss function at the same time.

## References


[1] Few-shot sound event detection. In *2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020 - Proceedings*, ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, pages 81–85. Institute of Electrical and Electronics Engineers Inc., May 2020. 1, 2

[2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. 1, 2

[3] https://dcase.community/challenge2021/task-few-shot-bioacoustic-event-detection results. 4

[4] https://github.com/c4dm/dcase-few-shot bioacoustic. 3

[5] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951. 2

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 2, 4

[7] Pau Rodríguez, Issam H. Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. *CoRR*, abs/2003.04151, 2020. 4

[8] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 2, 4

[9] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. *CoRR*, abs/1711.06025, 2017. 2

[10] Tiantian Tang, Yunhao Liang, and Yanhua Long. Two improved architectures based on prototype network for few-shot bioacoustic event detection. Technical report, DCASE2021 Challenge, June 2021. 4

[11] Yuxuan Wang, Pascal Getreuer, Thad Hughes, Richard F. Lyon, and Rif A. Saurous. Trainable frontend for robust and far-field keyword spotting. *CoRR*, abs/1607.05666, 2016. 3

[12] Dongchao Yang, Helin Wang, Zhongjie Ye, and Yuexian Zou. Few-shot bioacoustic event detection = a good transductive inference is all you need. Technical report, DCASE2021 Challenge, June 2021. 2, 4