applied sciences

MDPI

*Article*

# A Task-Adaptive Parameter Transformation Scheme for Model-Agnostic-Meta-Learning-Based Few-Shot Animal Sound Classification

**Jaeuk Moon, Eunbeen Kim, Junha Hwang and Eenjun Hwang ***

School of Electrical Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Republic of Korea; jaewookmo@korea.ac.kr (J.M.); gichanac@korea.ac.kr (E.K.); jhk97@korea.ac.kr (J.H.)
* Correspondence: ehwang04@korea.ac.kr; Tel.: +82-2-3290-3256

**Abstract:** Deep learning models that require vast amounts of training data struggle to achieve good animal sound classification (ASC) performance. Among recent few-shot ASC methods to address the data shortage problem regarding animals that are difficult to observe, model-agnostic meta-learning (MAML) has shown new possibilities by encoding common prior knowledge derived from different tasks into the model parameter initialization of target tasks. However, when the knowledge on animal sounds is difficult to generalize due to its diversity, MAML exhibits poor ASC performance due to the static initialization setting. In this paper, we propose a novel task-adaptive parameter transformation scheme called few-shot ASC. TAPT generates transformation variables while learning common knowledge and uses the variables to make parameters specific to the target task. Owing to this transformation, TAPT can reduce overfitting and enhance adaptability, training speed, and performance in heterogeneous tasks compared to MAML. In experiments on two public datasets on the same backbone network, we show that TAPT outperforms the existing few-shot ASC schemes in terms of classification accuracy, and in particular a performance improvement of 20.32% compared to the state-of-the-art scheme. In addition, we show that TAPT is robust to hyperparameters and efficient for training.

**Keywords:** animal sound classification; bioacoustics classification; deep learning; few-shot learning; meta-learning; MAML

check for updates

**Citation:** Moon, J.; Kim, E.; Hwang, J.; Hwang, E. A Task-Adaptive Parameter Transformation Scheme for Model-Agnostic-Meta-Learning-Based Few-Shot Animal Sound Classification. *Appl. Sci.* **2024**, *14*, 1025. https://doi.org/10.3390/app14031025

Academic Editor: Jie Tian

Received: 10 November 2023
Revised: 15 January 2024
Accepted: 24 January 2024
Published: 25 January 2024

**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Animal sound classification (ASC) has emerged as a crucial tool in wildlife monitoring systems, as it can identify different animal species based on their unique sounds [1]. ASC is particularly useful when visual identification is challenging, such as regarding small, nocturnal, and camouflaged animals [2]. Recently, deep learning-based models, such as convolutional neural networks (CNNs), have demonstrated superior performance in ASC as well as in other signal processing applications [3,4].

However, supervised deep learning ASC models (DeepASC) require huge amounts of accurately labeled data, and acquiring such data is an expensive and time-intensive process [5]. Insufficient animal sound data for DeepASC may result in poor classification performance due to overfitting or generalization failure [6–8]. This lack of labeled data is particularly severe in ASCs due to (1) the diversity of species and sounds, (2) limited access to certain habitats, especially remote or protected areas where specific species reside, (3) the time-consuming labeling process of animal sounds, and (4) the availability of experts, especially for animals that are difficult to observe (e.g., rare or endangered species). In these cases, DeepASC may not work effectively.

To mitigate this problem, few-shot learning that involves learning new tasks from only a small number of samples has garnered considerable attention. Recently, meta-learning, or learning to learn, has been widely used as one of the noteworthy methodologies for

few-shot learning. Meta-learning allows deep learning-based classification models to learn common prior knowledge shared across different tasks. Starting from the common knowledge, the classification models can easily learn specific knowledge about unseen tasks even with limited data samples.

Model-agnostic meta-learning (MAML), one of the most successful meta-learning methods, embeds the common knowledge derived from various tasks into the parameter initialization of the model [9]. Pretrained parameter initialization serves as a good starting point to overcome the data shortage problem and achieve good performance. Due to these properties, many recent works such as few-shot image classification [9], anomaly detection [10], and influenza forecasting [11] have aimed to achieve better pretrained generalization with MAML.

MAML is particularly useful in ASC because it enables DeepASC to learn common knowledge across various ASC source tasks, effectively adapting to specific knowledge of the target ASC task. Here, the target task is a classification task for rare or obscure animal species for which obtaining large amounts of labeled data is difficult. However, if the knowledge of the source tasks is too diverse, MAML often fails to generalize their common knowledge. As a result, the prior knowledge encoded in this failed generalization may be useful in some tasks but not in other tasks. This can be overcome by accommodating both task-wide general and task-specific knowledge into MAML-initialized parameters.

To carry out that, in this study, we propose a novel task-adaptive parameter transformation (TAPT) scheme that directly transforms the initial parameters of DeepASC according to the suitability of each task during the meta-learning. For parameter transformation, we consider gradients of initial parameters of DeepASC as the suitability to the task and use them as input to a regression model to learn task-specific knowledge. Then, we generate transformation variables from the regression model as output while generalizing common knowledge across tasks. Finally, these variables are used to adapt the initial parameters of DeepASC to the target task. Unlike traditional MAML, where initialization remains static across tasks, TAPT dynamically transforms the initial parameters based on task-specific knowledge. This property makes TAPT more flexible than MAML and more effective in classifying sounds of diverse species. To prove the effectiveness of the proposed scheme, we compared it with other few-shot ASC schemes in terms of ASC accuracy. We also analyzed the robustness of TAPT to hyperparameters and the training efficiency of TAPT and original MAML through the convergence speed of training accuracy.

The contributions of this paper are summarized as follows:

- We propose a novel parameter transformation scheme based on MAML to enhance the classification accuracy of DeepASC models. To the best of our knowledge, this is the first effort to use task-adaptive MAML for few-shot ASC.
- TAPT utilizes gradients as task-specific knowledge and incorporates this knowledge to adaptively transform the initial parameters for each task. This corresponds to an adaptable and efficient learning process.
- Task-specific initial parameters can lead to faster convergence during the fine-tuning phase. This is because DeepASC starts fine-tuning from a point in the parameter space that is closer to the optimal solution for a given task.
- We show that the proposed scheme can outperform ProtoNet, a state-of-the-art (SOTA) few-shot ASC scheme, using the same backbone network through experiments on two public animal sound datasets.
- We demonstrate that the proposed scheme is robust to hyperparameters, and it can significantly reduce training time compared to MAML.

The rest of this paper is presented as follows: Section 2 introduces the literature review about deep learning-based ASC models and few-shot learning schemes. Section 3 describes the proposed scheme in detail, and Section 4 presents the experimental setup and results; Section 5 provides conclusions along with future work.

## 2. Related Works

In this section, we first present a brief literature review on deep learning-based ASC, and then introduce several few-shot learning schemes to overcome the data shortage situation in ASC.

First of all, Şaşmaz et al. proposed a deep learning-based framework for classifying the sounds of various animal species, such as birds, cats, and dogs [12]. They collected 875 sound samples from 10 different animal species from an online sound source site and preprocessed animal audio data into mel-spectrograms. They then constructed three convolution layers with a max pooling layer and trained this model to classify target animal species.

Xie et al. proposed a bird sound classification structure that can incorporate acoustic features, visual features, and generalized features from a deep learning model; the first two features were obtained using traditional classifiers K-nearest neighbor and random forest, respectively, and the last generalized features were extracted from a three-layer CNN model [3]. Finally, the bird species is identified by incorporating these three features using a late fusion technique.

On the other hand, Zhang et al. proposed a method that can achieve outstanding bird sound classification accuracy based on deep CNNs (DCNNs) [4,13]. Specifically, they calculated spectrograms of the short-time Fourier transform, Mel-frequency transform, and Chirplet transform for animal sounds, constructed individual DCNN models for each spectrogram, and predicted bird species by combining the features from the DCNN models. Furthermore, they used a transfer-learning (TL) scheme to reduce the number of trainable parameters of the fusion model.

Liao et al. proposed a domestic pig sound classification model called TransformerCNN [14]. The model consists of two network modules, CNN and Transformer, in a parallel structure [15]. They decided that the spatial features extracted using a 4conv CNN (CNN with four convolutional layers) were not sufficient for ASC, and added sequential coding of the Transformer module to capture global features from the input spectrogram. The parallel structure model of the two modules effectively extracted richer information from different signals than a single structure model, and showed excellent performance in pig sound recognition.

The aforementioned deep learning-based ASC models usually require a large amount of labeled training data to achieve good performance. In addition, these models can only classify the animal species they are trained on. However, in the case of animals that are difficult to observe, the few-shot learning scheme, which learns how to classify animal species with a few data samples, has attracted great attention in situations of data shortage. For instance, Shi et al. proposed a few-shot acoustic event detection scheme based on three supervised learning schemes and three meta-learning schemes [16]: MetaOptNet [9], MAML [17], and Prototypical Networks (ProtoNet). For the Audioset [18] dataset containing music and animal sounds, they compared those learning schemes in 5-way 1-shot and 5-way 5-shot settings. As a result, all meta-learning schemes outperformed the supervised learning schemes, and ProtoNet was the best among the meta-learning schemes.

Meanwhile, many high-ranked methods for few-shot bioacoustics event detection presented in the DCASE 2022 challenge [19] used ProtoNet as a learning scheme for training CNN-based ASC models. Here, bioacoustics event detection refers to locating animal sounds in audio recordings and classifying animal species. Although bioacoustics event detection is slightly different from ASC, this indicates that ProtoNet is considered an effective model for ASC.

## 3. Proposed Scheme

In this section, we first describe the data collection and preprocessing process. Next, we present an MAML-based few-shot ASC process. Finally, the learning process of the proposed scheme is presented in detail. Figure 1 and Algorithm 1 show the overall architecture of TAPT and the meta-training process of TAPT, respectively.

**Algorithm 1** Meta-training process of TAPT

**Input:** Original waveform data $D_w$, learning rates $\alpha$, $\eta$
**Output:** Pretrained parameters $\theta$ from a ASC model $f_\theta$

$D_w\prime \leftarrow STFT(Padding_{1\,\mathrm{sec}}(Sampling_{16\mathrm{kHz}}(D_w)))$
Construct a set of meta-training tasks $S_{task} = \{T_1, \ldots, T_N\}$ ($T_i$: $m$-way $k$-shot task) from the meta-training set $M_{train}$
Randomly initialize $\theta$, $\varnothing_\gamma$, $\varnothing_\beta$
Let $\theta = \left\{\theta^j\right\}^{j=1\ldots l}$ where $j$ is the layer index and $l$ is the number of layers of a network
**while** meta-learning epochs **do**
  Sample a batch from task $T_i$
  **for** each task $T_i$ **do**
    Sample data samples ($D_{T_i}$ , $D'_{T_i}$) from $T_i$
    Compute $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{D_{T_i}}(f_\theta)$
    Generate layer-wise transformation parameters ($\gamma$, $\beta$)
    $\left\{\gamma_i{}^j\right\}^{j=1\ldots l} = g_{\varnothing_\omega}(\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{D_{T_i}}(f_\theta))$
    $\left\{\beta_i{}^j\right\}^{j=1\ldots l} = g_{\varnothing_\beta}(\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{D_{T_i}}(f_\theta))$
    Compute transformed initial parameters:

$$\bar{\theta}_i^j = \gamma_i^j \theta^j + \beta_i^j$$

    Initialize $\theta_i' = \left\{\bar{\theta}_i^j\right\}^{j=1\ldots l}$
    **for** number of inner-loop updates **do**
      Compute $\mathcal{L}_{\mathcal{T}_i}^{D_{T_i}}\left(f_{\theta_i'}\right)$
      Perform gradient descent to compute transformed parameters:

$$\theta\prime_i = \theta\prime_i - \alpha \nabla_{\theta'_i} \mathcal{L}_{\mathcal{T}_i}^{D_{T_i}}\left(f_{\theta'_i}\right)$$

    **end for**
    Compute $\mathcal{L}_{\mathcal{T}_i}^{D'_{T_i}}\left(f_{\theta_i'}\right)$
  **end for**
  Perform gradient descent to update weights:

$$\left(\theta, \varnothing_\gamma, \varnothing_\beta\right) \leftarrow \left(\theta, \varnothing_\gamma, \varnothing_\beta\right) - \eta \nabla_{(\theta, \varnothing_\gamma, \varnothing_\beta)} \sum_{i=1}^{N} \mathcal{L}_{\mathcal{T}_i}^{D'_{T_i}}\left(f_{\theta_i'}\right)$$

**end while**

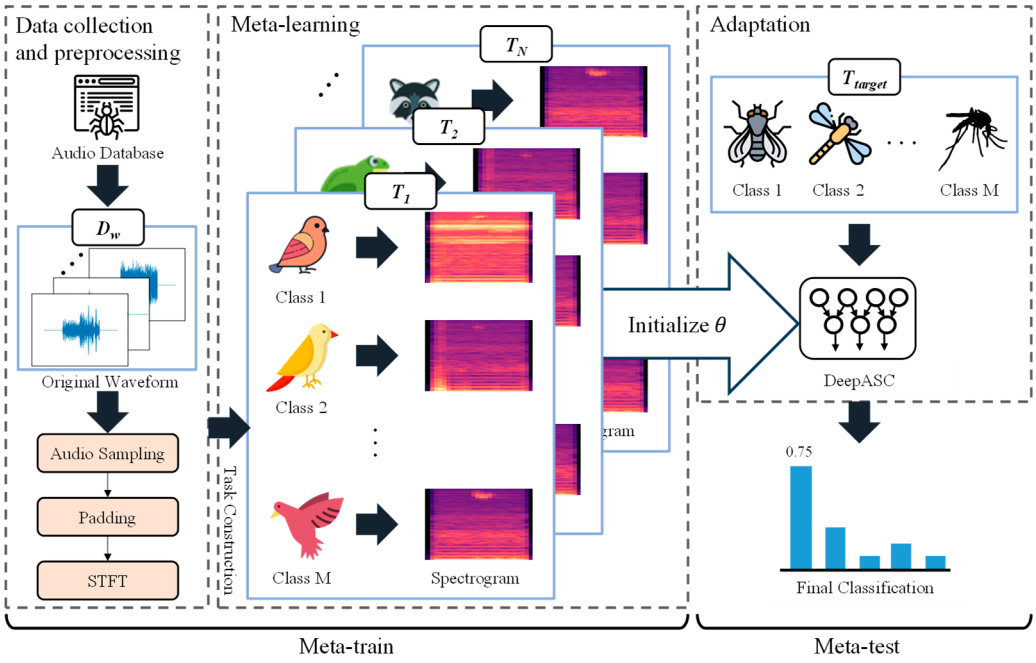



**Figure 1.** Overall structure of MAML-based ASC.

### 3.1. Data Collection and Preprocessing

We first collect original waveform data $D_w$ from animal sound databases or bioacoustics sensors. To use animal sounds as input to DeepASC, preprocessing into a spectrogram is essential. Compared to the original waveform, a spectrogram contains a frequency–time representation (i.e., 2D visual representation) that shows how the frequency content of a sound signal changes over time. The preprocessing part (Line 1 in Algorithm 1) is organized as follows: (i) all sound segments in the datasets are sampled at a sampling rate of 16 kHz and (ii) padded to a length of one second. (iii) Short-time Fourier transformation (STFT) is conducted on the raw waveform with an FFT size of 256, a window size of 128, and a hop size of 128 to obtain 128 $\times$ 87 spectrograms that constitute $D_w$'.

### 3.2. MAML-Based ASC

In this section, we formulate the MAML algorithm for ASC. First, we divide whole animal classes into meta-training set $M_{train}$ and meta-test set $M_{test}$. The meta-training and meta-test processes use $M_{train}$ and $M_{test}$, respectively. Then, we sample the ASC task from both $M_{train}$ and $M_{test}$. Here, "task" represents ASC for $m$ animal sound classes with $k$ samples (i.e., $m$-way $k$-shot). From $M_{train}$, we sample $S_{task}$, which represents a set of meta-training tasks consisting of $N$ different tasks, $T_1, \ldots, T_N$ (Line 2). From the $M_{test}$, we sample $T_{target}$, which implies the target ASC task.

In the meta-learning stage, MAML encodes common prior knowledge derived from the meta-training task set $S_{task}$ into the initial parameters $\theta$ of DeepASC $f_\theta$. This initialization can be used as a good starting point and allows $f_\theta$ to quickly adapt to an unseen target task, $T_{target}$, in the adaption stage. In the original MAML, this stage consists of two loops, an inner loop and an outer loop. In the inner loop (Line 15–17), the weights of $f_\theta$ are adapted to $T_i$ for a small number ($k$) of animal sound samples, $D_{T_i}$ (support set from $T_i$), and a loss function $\mathcal{L}_{T_i}$ as follows:

$$\theta\prime_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{D_{T_i}}(f_\theta) \tag{1}$$

In the outer loop (Line 21), the model $f_\theta$ is evaluated for unseen-animal sound samples, $D\prime_{T_i}$ (query set from $T_i$), to provide feedback on the generalization performance. This feedback is used to update the initial $\theta$ for all the $S_{task}$ tasks to achieve a better generalization of the common knowledge as follows:

$$\theta \leftarrow \theta - \eta \nabla_\theta \sum_{i=1}^{N} \mathcal{L}_{\mathcal{T}_i}^{D'_{T_i}}\left(f_{\theta_i'}\right) \tag{2}$$

After the meta-learning stage, DeepASC $f_\theta$ learns the specific knowledge of the target classification task $T_{target}$ during the adaptation stage. Here, the classes used in this stage are unseen in the meta-learning stage, and MAML allows the model to quickly adapt to the specific knowledge, starting from the initial $\theta$. In this stage, fine-tuning is conducted according to Equation (1) using a small number of samples, $D_{T_{target}}$ (support set), obtained from the target task to make $f_\theta$ into $f_{\theta_{target}}$, a model trained with the specific knowledge of the target task. After the adaptation stage, $f_{\theta_{target}}$ is used to classify the new samples (query set) obtained from the target task.

### 3.3. Task-Adaptive Parameter Transformation

In the meta-learning stage, we consider $N$ tasks, each consisting of $n$ animal sound samples with different characteristics (knowledge). When these tasks have very diverse knowledge, MAML may have difficulty in generalizing them into common knowledge. Initial parameters set through such generalization may result in poor ASC performance. To address this, we convert the initial parameters $\theta$ directly to task-specific parameters $\overline{\theta}$ according to task suitability. The task-specific parameters can provide better ASC performance compared to the original MAML by considering task-specific knowledge as well as common knowledge.

Two essential factors to be determined in the process are (i) the suitability of the initial parameters $\theta$ to each task and (ii) the amount of transformation according to the suitability. In order to assess the suitability of $\theta$ to the $i$-th task $T_i$, we use gradients, $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}^{D_{T_i}}(f_\theta)$ (Line 9). Although gradients are typically utilized to update parameters via a gradient descent, they also contain information about optimization and parameter quality. Thus, gradients can also be used to represent the meta-information (i.e., suitability) of parameters to a task [20,21].

In order to obtain an optimal set of transformations of the initial parameters, we construct regression models. The new models $g_{\varnothing_\gamma}$ and $g_{\varnothing_\beta}$ (parameterized by $\varnothing_\gamma$ and $\varnothing_\beta$, respectively) take the gradients of $T_i$ as inputs and generate transformation variables $\gamma_i$ and $\beta_i$. These variables are used for transforming the task-wide initial parameters $\theta$ into task-specific parameters $\overline{\theta}_i$ of $T_i$ (Line 10–13). Here, the two models $g_{\varnothing_\gamma}$ and $g_{\varnothing_\beta}$ are two-layer multilayer perceptrons, with ReLU and tanh as the activation functions at the end, respectively. Further, $\gamma_i = \left\{\omega_i^j\right\}$ and $\beta_i = \left\{\beta_i^j\right\}$ are sets of the layer-wise transformation variables of $T_i$ for the $j$-th layer of the DeepASC model parameter $\theta_i$.

In the adaptation stage, we transform the task-wise parameters $\theta$, which are adaptive to $T_{target}$, with the gradients $\nabla_\theta \mathcal{L}_{\mathcal{T}_{target}}^{D_{T_{target}}}(f_\theta)$ and conduct fine-tuning using Equation (1).

## 4. Experiments

### 4.1. Datasets

To evaluate the effectiveness of TAPT, we performed various experiments using two public animal sound datasets, BirdVox-14SD and ANAFCC [22], with significantly different class distributions. BirdVox-14SD contains 6600 h of audio derived from 37 classes of avian animals (e.g., birds and insects) and collected from ten autonomous recording units located in Ithaca, New York, USA. Among these classes, we excluded 16 classes that contained audio of unknown animal species or audio of several species, and used the remaining 21 classes for the experiments. ANAFCC contains short audio waveforms of bird flight calls derived from 27 classes. In this case, we excluded 12 classes for the same reason as in BirdVox-14SD and used the remaining 16 classes.

The dataset is divided into 2 subsets: the meta-training set $M_{train}$ (from which $S_{task}$ is extracted) and the meta-test set $M_{test}$ (from which $T_{target}$ is extracted). We first sorted the classes of the datasets according to the number of data samples. Then, to represent the data shortage situation, we used the classes with a small number of samples as the meta-test set and the remaining classes as the $M_{train}$. As a result, 15 classes and 6 classes for BirdVox-14SD and 10 classes and 5 classes for ANAFCC were used as $M_{train}$ and $M_{test}$, respectively. Table 1 shows the number of classes and samples in $M_{train}$ and $M_{test}$ of each dataset; Table 2 presents classes in the datasets and their corresponding animal species; and Figure 2 illustrates the number of data samples from each class in the datasets.

**Table 1.** Number of classes and samples in meta-training and meta-test for each dataset.

| Dataset | $M_{train}$ | | $M_{test}$ | |
|---|---|---|---|---|
| BirdVox-14SD | # of classes | # of samples | # of classes | # of samples |
| | 15 | 4644 | 6 | 131 |
| ANAFCC | # of classes | # of samples | # of classes | # of samples |
| | 10 | 40821 | 5 | 2283 |

**Table 2.** Classes in the datasets and their corresponding animal species.

| Class | Animal Species |
|---|---|
| 1.1.1 | American tree sparrows |
| 1.1.2 | Chipping sparrow |
| 1.1.3 | Savannah sparrow |
| 1.1.4 | White-throated sparrow |
| 1.2.1 | Rose-breasted grosbeak |
| 1.3.1 | Gray-cheeked thrush |
| 1.3.2 | Swainson's thrush |
| 1.4.1 | American redstart |
| 1.4.2 | Bay-breasted warbler |
| 1.4.3 | Black-throated blue warbler |
| 1.4.4 | Canada warbler |
| 1.4.5 | Common yellowthroat |
| 1.4.6 | Mourning warbler |
| 1.4.7 | Ovenbird |
| 0.X.X | Bugs and insects |

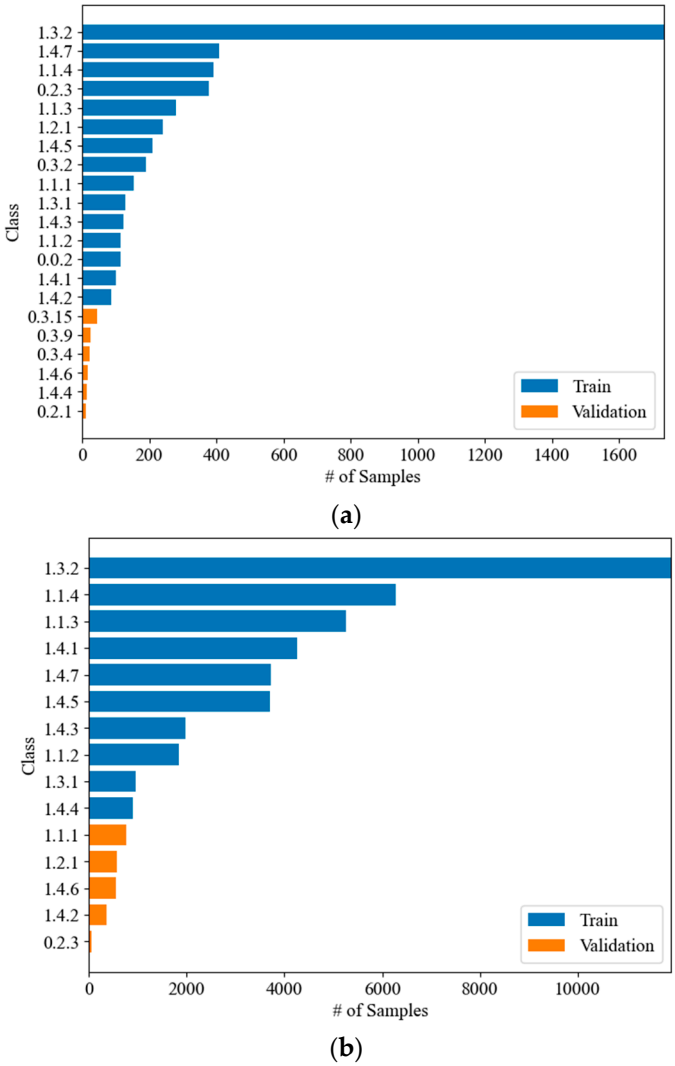



**Figure 2.** Number of data samples from each class in the datasets. (**a**) BirdVox-14SD; (**b**) ANAFCC.

### *4.2. Experimental Settings*

For ASC comparison, we considered five learning schemes, CNN, TL, MAML, ProtoNet, and TAPT, and two backbone networks (DeepASC), 4conv and VGG11. The 4conv

network consists of four layers, each of which contains thirty-two 3 × 3 convolutional filters, a batch normalization function, a ReLU activation function, and a 2 × 2 max pooling function. On the other hand, VGG11 consists of eleven layers: eight 3 × 3 convolutional layers with max pooling function and three fully connected layers [23].

All learning schemes except CNN pretrained the backbone network using data obtained from the meta-training set $S_{task}$ according to the training process of each scheme and conducted fine-tuning using data obtained from $T_{target}$. In the case of CNN, the backbone network was trained using the animal sound data derived from the target task $T_{target}$.

For a fair performance comparison, all learning schemes utilized an identical preprocessing process and hyperparameter values. For example, the meta batch size, learning rates ($\alpha$ and $\eta$), and number of inner-loop updates were 4, 0.001, and 5, respectively. Here, we used the meta batch size and number of inner-loop updates from [5] and set the learning rate empirically. Furthermore, we set the epoch of meta-learning (pretraining in the case of TL) and adaptation to 50 and 10, respectively, while each learning scheme can be stopped early during meta-learning.

### *4.3. Few-Shot ASC Performance*

Few-shot ASC was performed in typical settings such as 5-way 1-shot classification and 5-way 5-shot classification. As an evaluation metric, we used accuracy, which represents the ratio of the number of correct samples to the total number of samples.

Tables 3 and 4 present the accuracy comparison results of the learning schemes for the BirdVox-14SD dataset and the ANAFCC dataset, respectively. Here, the accuracy values represent the mean and 95% confidence interval values for ten repeated experiments. The tables show that TAPT achieves the best accuracy compared to other learning schemes.

**Table 3.** Accuracy comparison of 5-way ASC on the BirdVox-14SD dataset (Bold and underlined values indicate the best and second-best accuracies, respectively).

| Learning Scheme | 4conv | | VGG11 | |
|---|---|---|---|---|
| | 1-Shot | 5-Shot | 1-Shot | 5-Shot |
| CNN | 27.86 ± 0.28% | 36.24 ± 0.12% | 31.23 ± 0.27% | 40.48 ± 0.11% |
| TL | 32.39 ± 1.95% | 49.19 ± 1.91% | 45.68 ± 1.37% | 66.32 ± 1.14% |
| MAML | 45.83 ± 1.01% | 56.24 ± 0.56% | <u>59.23 ± 0.93%</u> | 70.08 ± 0.87% |
| ProtoNet | <u>57.45 ± 1.02%</u> | <u>76.08 ± 0.67%</u> | 58.63 ± 1.18% | <u>70.15 ± 1.15%</u> |
| TAPT | **64.45 ± 0.49%** | **83.48 ± 0.43%** | **66.85 ± 0.79%** | **84.41 ± 0.42%** |

**Table 4.** Accuracy comparison of 5-way ASC on the ANAFCC dataset (Bold and underlined values indicate the best and second-best accuracies, respectively).

| Learning Scheme | 4conv | | VGG11 | |
|---|---|---|---|---|
| | 1-Shot | 5-Shot | 1-Shot | 5-Shot |
| CNN | 41.60 ± 0.11% | 46.05 ± 0.13% | 43.86 ± 0.08% | 61.79 ± 0.07% |
| TL | 44.19 ± 0.43% | 49.89 ± 0.33% | 40.26 ± 0.60% | 75.68 ± 0.55% |
| MAML | 55.32 ± 0.23% | 70.25 ± 0.21% | <u>65.19 ± 0.18%</u> | <u>75.97 ± 0.89%</u> |
| ProtoNet | <u>59.03 ± 0.29%</u> | <u>72.44 ± 0.30%</u> | 59.90 ± 0.40% | 72.41 ± 0.38% |
| TAPT | **64.16 ± 0.25%** | **77.61 ± 0.63%** | **70.86 ± 0.43%** | **81.12 ± 0.25%** |

In addition, Figures 3 and 4 compare the accuracy of the learning schemes on the two backbone networks for the BirdVox-14SD dataset and the ANAFCC dataset, respectively. From the figures, it can be seen that all learning schemes show better accuracy on VGG11 in most cases due to the depth and number of parameters.

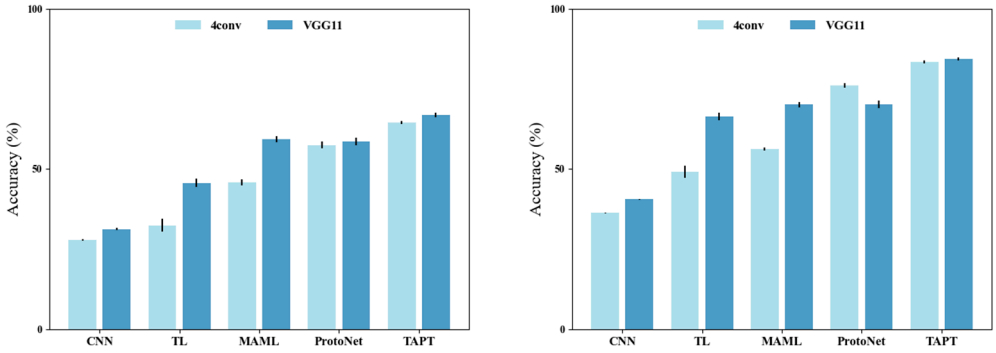



**Figure 3.** Accuracy comparison of the ASC schemes on the BirdVox-14SD dataset in (**a**) 1-shot and (**b**) 5-shot settings.

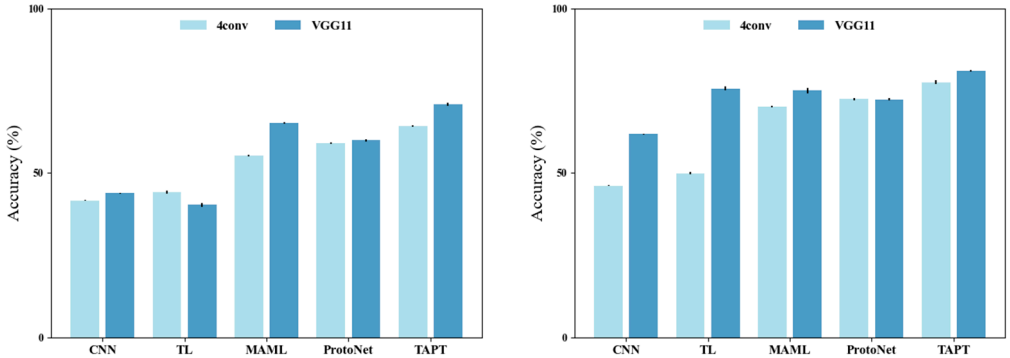



**Figure 4.** Accuracy comparison of the ASC schemes on the ANAFCC dataset in (**a**) 1-shot and (**b**) 5-shot settings.

Furthermore, in the tables, few-shot learning schemes such as MAML and TL showed better ASC performance than CNN because they were trained with $S_{task}$ in addition to $T_{target}$.

Among the few-shot learning schemes, MAML outperformed TL. One advantage of MAML over TL in few-shot ASC is generalization. As TL lacks the generalization process of knowledge during pretraining of the source task, its adaptability to new tasks in a few-shot setting is limited. In contrast, MAML updates the parameters of DeepASC in a way that generalizes the knowledge of the source task, resulting in better ASC accuracy in all cases. Despite MAML's superior performance, ProtoNet showed better classification performance than MAML in most cases. We found several advantages of ProtoNet compared to MAML in the context of few-shot ASC. First, ProtoNet computes the prototype of each class in a feature space, allowing for a more efficient representation of each class and making it easier to classify new classes based on their class prototypes. MAML, while powerful, learns an initialization that can quickly adapt to new tasks. In some cases, the absence of explicit prototypes might lead to less efficient representations, especially in limited labeled data. This simplicity and efficiency make ProtoNet SOTA in few-shot ASC and it is used in various challenges like few-shot animal sound event detection (DCASE). However, we found that ProtoNet might not capture fine-grained task-specific knowledge because ProtoNet's nearest-neighbor classification is based on class prototypes.

To address this, we propose a novel task-adaptive MAML that transforms parameter initialization according to task-specific knowledge for more accurate adaptation to new tasks. This fine-grained task-specific learning improves classification accuracy, especially for classes with subtle differences. To sum up, by customizing parameter initialization according to task-specific knowledge, TAPT addresses key limitations of MAML and

ProtoNet, offering enhanced adaptability, fine-grained task-specific learning, efficient knowledge transfer, improved generalization, and balancing between complexity and flexibility. Due to these properties, quantitative results demonstrate that TAPT significantly outperforms comparison schemes in all settings.

Overall, TAPT performed the best in all cases on both datasets, achieving relative gains of up to 20.32% and 8.69% on the BirdVox-14SD and ANAFCC datasets, respectively, when compared to the second-best scheme. In summary, the original MAML showed serious limitations in ASC, and TAPT was able to achieve outstanding ASC performance by transforming the initial parameters of DeepASC more adaptively than MAML.

### *4.4. Sensitivity Analysis and Ablation Study*

In this experiment, we performed a sensitivity analysis of TAPT to evaluate the effect of the number of inner-loop updates on ASC performance, and the results are shown in Table 5 and Figure 5. According to the table, TAPT is less affected by the number of inner loops in ASC accuracy compared to MAML. Also, the lowest accuracy of TAPT is better than that of all existing few-shot ASC schemes using five inner-loop updates, as shown in Table 3. This indicates that the proposed scheme is more robust to the hyperparameters compared to other learning schemes.

**Table 5.** Accuracy of 5-way 5-shot ASC with number of inner-loop update steps on the BirdVox-14SD dataset.

| Inner-Loop Update Steps | MAML | TAPT |
|---|---|---|
| 1 | 49.93 ± 0.32% | 82.16 ± 0.47% |
| 2 | 48.63 ± 0.51% | 80.85 ± 0.49% |
| 3 | 52.14 ± 0.49% | 81.61 ± 0.46% |
| 4 | 55.94 ± 0.51% | 82.55 ± 0.41% |
| 5 | 56.24 ± 0.56% | 83.48 ± 0.43% |

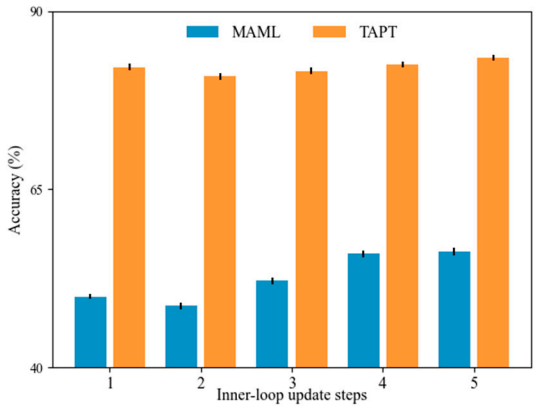


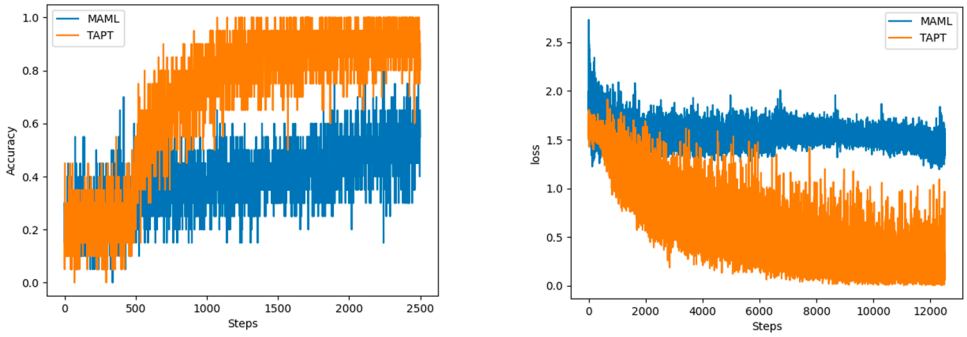

**Figure 5.** Comparison of 5-way 5-shot ASC accuracy of MAML and TAPT over inner-loop update steps using the BirdVox-14SD dataset.

Next, as an ablation study, we compare the ASC performance of MAML and TAPT according to the number of inner-loop updates. The comparison between MAML and TAPT in Tables 3 and 4 might be unfair, because TAPT adjusts its parameters once more for initialization before the inner-loop update. However, even TAPT with one inner-loop update provides 46.08% better ASC performance than the original MAML with five inner-loop updates, as shown in Table 5. This is attributed to the task-adaptive transformation of TAPT that allows DeepASC to quickly adapt to unseen tasks.

### *4.5. Comparison of Training Accuracy*

In the last experiment, we compare the training accuracy of MAML and TAPT. Figure 6a,b present their 5-way 5-shot training accuracy and loss, respectively, according to

the number of inner-loop update steps contained in the training epoch for the ANAFCC dataset when using 4conv as a backbone. Figure 6 shows that compared to MAML, TAPT's training accuracy and training loss quickly converge to 1 and 0, respectively. This means that the parameter transformation of TAPT enables DeepASC to quickly adapt to unseen animal sound data. As a result, the epochs and training time required to train TAPT are much less compared to MAML.




(**a**) (**b**)

**Figure 6.** Comparison of training processes between TAPT and MAML according to training steps. (**a**) Training accuracy; (**b**) training loss.

## 5. Conclusions

In this paper, we proposed TAPT, a novel MAML-based task-adaptive parameter transformation scheme that can alleviate the data shortage problem in ASC. TAPT generates transformation variables while generalizing common knowledge and uses them to adjust the parameters of each specific classification task. To evaluate the effectiveness of the proposed scheme, we conducted extensive experiments for five different learning schemes using two public datasets and two backbone networks. In the experimental results, TAPT showed better performance than other few-shot ASC schemes, achieving up to 20.32% improvement in ASC accuracy compared to the SOTA scheme. In addition, a sensitivity analysis confirmed that TAPT is robust to the number of inner-loop updates, and an ablation study proved that the ASC accuracy improvement in TAPT results from the proposed parameter transformation. Finally, training accuracy comparison demonstrated that TAPT learns to classify unseen tasks more efficiently than MAML.

In the future, based on our few-shot learning scheme, we plan to make software that can be mounted on the audio sensor equipment for animal species classification. In addition, we will consider a hybrid MAML scheme that can incorporate features from the spectrogram as well as the waveform.

**Author Contributions:** Conceptualization, J.M.; methodology, J.M.; validation, J.M.; formal analysis, J.M.; investigation, E.K. and J.H.; data curation, E.K. and J.H.; visualization, J.M.; writing—original draft preparation, J.M.; writing—review and editing, E.H.; supervision, E.H.; project administration, E.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2023-00262750, Development of Automated Surveillance System for Vector Mosquitoes (Japanese encephalitis, Malaria) using AI-Based Sound Recognition Technology).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are openly available from Zenodo at "https://doi.org/10.5281/zenodo.3666782" (accessed on 10 November 2023) and "https://doi.org/10.5281/zenodo.3667094" (accessed on 10 November 2023).

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References


1. Potamitis, I.; Ntalampiras, S.; Jahn, O.; Riede, K. Automatic bird sound detection in long real-field recordings: Applications and tools. *Appl. Acoust.* **2014**, *80*, 1–9. [CrossRef]
2. Kim, E.; Moon, J.; Shim, J.; Hwang, E. DualDiscWaveGAN-Based Data Augmentation Scheme for Animal Sound Classification. *Sensors* **2023**, *23*, 2024. [CrossRef] [PubMed]
3. Xie, J.; Zhu, M. Handcrafted features and late fusion with deep learning for bird sound classification. *Ecol. Inform.* **2019**, *52*, 74–81. [CrossRef]
4. Zhang, F.; Zhang, L.; Chen, H.; Xie, J. Bird Species Identification Using Spectrogram Based on Multi-Channel Fusion of DCNNs. *Entropy* **2021**, *23*, 1507. [CrossRef] [PubMed]
5. Baik, S.; Hong, S.; Lee, K.M. Learning to forget for meta-learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 14–19 June 2020; pp. 2379–2387.
6. Xiao, X.; Mo, H.; Zhang, Y.; Shan, G. Meta-ANN–A dynamic artificial neural network refined by meta-learning for Short-Term Load Forecasting. *Energy* **2022**, *246*, e123418. [CrossRef]
7. Zhang, S.; Ye, F.; Wang, B.; Habetler, T.G. Few-shot bearing anomaly detection via model-agnostic meta-learning. In Proceedings of the 23rd IEEE International Conference Electrical Machines and Systems, Hamamatsu, Japan, 24–27 November 2020; pp. 1341–1346. [CrossRef]
8. Deng, S.; Wang, S.; Rangwala, H.; Wang, L.; Ning, Y. Cola-GNN: Cross-location Attention based Graph Neural Networks for Long-term ILI Prediction. In Proceedings of the 29th ACM International Conference Information and Knowledge Management, New York, NY, USA, 19–23 October 2020; pp. 245–254. [CrossRef]
9. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
10. Moon, J.; Noh, Y.; Jung, S.; Lee, J.; Hwang, E. Anomaly detection using a model-agnostic meta-learning-based variational auto-encoder for facility management. *J. Build. Eng.* **2023**, *68*, 106099. [CrossRef]
11. Moon, J.; Noh, Y.; Park, S.; Hwang, E. Model-agnostic meta-learning-based region-adaptive parameter adjustment scheme for influenza forecasting. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 175–184. [CrossRef]
12. Şaşmaz, E.; Tek, F.B. Animal sound classification using a convolutional neural network. In Proceedings of the 2018 3rd International Conference on Computer Science and Engineering, Sarajevo, Bosnia and Herzegovina, 20–23 September 2018; pp. 625–629. [CrossRef]
13. Merchan, F.; Guerra, A.; Poveda, H.; Guzmán, H.M.; Sanchez-Galan, J.E. Bioacoustic classification of Antillean manatee vocalization spectrograms using deep convolutional neural networks. *Appl. Sci.* **2020**, *10*, 3286. [CrossRef]
14. Liao, J.; Li, H.; Feng, A.; Wu, X.; Luo, Y.; Duan, X.; Ni, M.; Li, J. Domestic pig sound classification based on TransformerCNN. *Appl. Intell.* **2022**, *53*, 4907–4923. [CrossRef]
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
16. Shi, B.; Sun, M.; Puvvada, K.C.; Kao, C.C.; Matsoukas, S.; Wang, C. Few-shot acoustic event detection via meta learning. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Barcelona, Spain, 4–8 May 2020; pp. 76–80. [CrossRef]
17. Lee, K.; Maji, S.; Ravichandran, A.; Soatto, S. Meta-learning with differentiable convex optimization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 29 October–1 November 2019; pp. 10657–10665.
18. Gemmeke, J.F.; Ellis, D.P.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R.C.; Plakal, M.; Ritter, M. Audio set: An ontology and human-labeled dataset for audio events. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, 5–9 March 2017; pp. 776–780. [CrossRef]
19. Nanni, L.; Costa, Y.M.; Aguiar, R.L.; Mangolin, R.B.; Brahnam, S.; Silla, C.N. Ensemble of convolutional neural networks to improve animal audio classification. *EURASIP J. Audio Speech Music Process.* **2020**, *2020*, 8. [CrossRef]
20. Younger, A.S.; Conwell, P.R.; Cotter, N.E. Fixed-weight on-line learning. *IEEE Trans. Neural Netw.* **1999**, *10*, 272–283. [CrossRef] [PubMed]
21. Mitchell, T.M.; Thrun, S.B. Explanation-based neural network learning for robot control. *Adv. Neural Inf. Process. Syst.* **1992**, *5*, 287–294.
22. Cramer, A.L.; Lostanlen, V.; Farnsworth, A.; Salamon, J.; Bello, J.P. Chirping up the right tree: Incorporating biological taxonomies into deep bioacoustic classifiers. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 901–905. [CrossRef]
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.