ACOUSTIC SCENE CLASSIFICATION ACROSS MULTIPLE DEVICES THROUGH INCREMENTAL LEARNING OF DEVICE-SPECIFIC DOMAINS

Manjunath Mulimani, Annamaria Mesaros

Signal Processing Research Center, Tampere University, Tampere, Finland
{manjunath.mulimani, annamaria.mesaros}@tuni.fi

ABSTRACT

In this paper, we propose using a domain-incremental learning approach for coping with different devices in acoustic scene classification. While the typical way to handle mismatched training data is through domain adaptation or specific regularization techniques, incremental learning offers a different approach. With this technique, it is possible to learn the characteristics of new devices on-the-go, adding to a previously trained model. This also means that new device data can be introduced at any time, without a need to retrain the original model. In terms of incremental learning, we propose a combination of domain-specific Low-Rank Adaptation (LoRA) parameters and running statistics of Batch Normalization (BN) layers. LoRA adds low-rank decomposition matrices to a convolutional layer with a few trainable parameters for each new device, while domain-specific BN is used to boost performance. Experiments are conducted on the TAU Urban Acoustic Scenes 2020 Mobile development dataset, containing 9 different devices; we train the system using the 40h of data available for the main device, and incrementally learn the domains of the other 8 devices based on 3h of data available for each. We show that the proposed approach outperforms other fine-tuning-based methods, and is outperformed only by joint learning with all data from all devices.

Index Terms— Domain-incremental learning, Low-Rank Adaptation, Batch Normalization, acoustic scene classification, mismatched devices

1. INTRODUCTION

Deep learning models have recently shown impressive results for acoustic scene classification (ASC) tasks from in-domain static data. However, in realistic scenarios, new data comes in sequentially. This new data may be from a different domain than the data used to optimize the model. Incremental or continuous learning of such a sequence of mismatched domains (i.e., locations, devices, or other acoustic conditions) deteriorates the model performance on previously learned domains when learning a new one, which means catastrophic forgetting [1] occurs in the absence of the previous domain's data. Mismatched conditions in continuously evolving domains introduce domain shift or bias in the feature distribution, which is the main reason for performance degradation.

In this work, we propose to use the domain-incremental learning (DIL) [2] approach for learning ASC tasks from different domains (devices) without forgetting the acoustic scenes from previously seen domains.

This work was supported by Jane and Aatos Erkko Foundation grant 230048 "Continual learning of sounds with deep neural networks". The authors wish to thank CSC-IT Centre of Science Ltd., Finland, for providing computational resources.

DIL was successfully applied to detect objects from road scenes in different locations [3] and in different weather conditions [2] for images, and acoustic scenes from different locations [4]. We aim to develop a practical DIL model to effectively classify acoustic scenes from all recording devices seen so far by going through the stream of data only once, in online learning mode.

DIL is different than existing domain adaptation (DA) methods for ASC from different devices [5–7]. DA setup typically includes two domains: source and target. It transfers the knowledge from the source to the target domain and only focuses on the accuracy of the target domain. DA requires access to the data of the source domain to match the distribution with the target domain. In comparison to DA, the DIL setup includes multiple domains over time that the system needs to adapt to; it focuses on the overall accuracy of all the domains seen so far; takes additional measures to alleviate the forgetting; and typically does not have access to the previous domain's data.

Our previous work adapts the model for the new locations sequentially by updating only the running statistics i.e., running mean and variance of BN layers in an online domain incremental learning (ODIL) setup [4]. In this work, we propose to add Low-Rank Adaptation (LoRA) parameters to the convolutional layers of the model, and update only these LoRA parameters and running statistics of the BN layers to adapt to the incrementally occurring new devices for effective ASC. LoRA is a parameter-efficient fine-tuning (PEFT) method widely used as a fine-tuning strategy for transformer-based Large Language Models (LLMs) [8]. LoRA is also used with vision transformers for continual learning of images [9] and also applied to convolutional layers for DA [10] and segmentation [11] of images.

The use of LoRA with CNN-based models for ODIL in the context of audio devices is yet to be explored. Unlike conventional fine-tuning, in which all the parameters of the model are updated to adapt to a new domain, LoRA fixes the other parameters of the current model and only updates the trainable low-rank matrices on the new domain, sequentially. LoRA parameters are significantly less than the total parameters of the original model.

The main contributions of this work are as follows,

- We propose using LoRA parameters for ODIL to learn acoustic scenes incrementally from mismatched devices.
- We investigate the combination of LoRA and BN statistics in classifying acoustic scenes in both online and offline settings.
- We also investigate the ability of the proposed approach trained on a device with enough data to adapt to incoming mismatched devices with limited data. It verifies the suitability of LoRA in low-data scenarios.

The rest of the paper is organized as follows: Section 2 presents the notations, baselines, and the proposed LoRA and BN combination

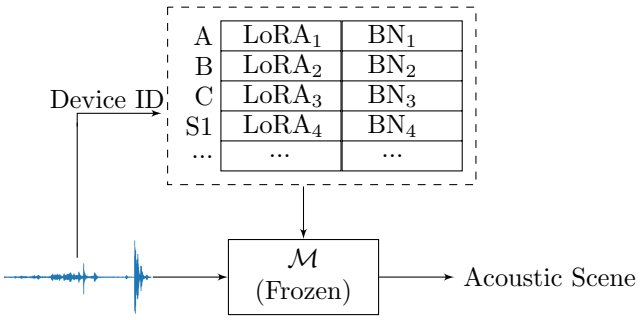


Figure 1: Overview of the proposed approach for incremental learning of acoustic scenes from different devices in sequence. Inputs to the model are the test sample and the device ID. The frozen model $\mathcal{M}$ uses domain-specific LoRA parameters and BN statistics to classify the acoustic scenes from a particular device such as A, B, C, S1, and so on.

for ODIL of acoustic scenes. Section 3 introduces the datasets, implementation details, and results. Finally, conclusions are given in Section 4.

2. INCREMENTAL LEARNING OF DEVICE DOMAINS

2.1. Incremental learning setup and notations

In our incremental learning setup, a sequence of ASC tasks is presented to the model; these tasks represent the datasets from different domains: $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_t$. The model learns each task, i.e., $\mathcal{D}_t$ in our case, at incremental time step t. A domain $\mathcal{D}_t$ is an acoustic scene dataset recorded with a particular device, composed of audio clips and corresponding class labels. All domains share the same classes. We aim to train a single-model $\mathcal{M}$ that learns to classify the same acoustic scenes when domain or data distribution changes. Initially, we train the $\mathcal{M}$ on a relatively larger dataset $\mathcal{D}_1$ offline and this model is a base model for incremental tasks. During the training of incremental tasks, $\mathcal{M}$ follows a realistic setting where it sees a stream of samples only once, online, and quickly adapts to the new domain on the fly, i.e., ODIL. More importantly, the performance of the $\mathcal{M}$ does not degrade on previous domains when it learns a new domain, unlike the domain adaptation case, in which the performance on the previous domain does not matter. Note that in this work we refer to $\mathcal{D}_t$ as task, domain, device, and dataset interchangeably.

2.2. Baselines

We construct a few standard baselines to compare with the proposed approach: (1) *Feature extraction (FE)*: the feature extractor component of the base model is frozen after learning $\mathcal{D}_1$. The classifier is updated in each incremental domain; (2) *Conventional Fine-tuning (FT)*: a model trained on the previous domain is fine-tuned on the new domain at each incremental time step with all its parameters. The model is being trained incrementally; (3) *Disjoint*: a base model is trained separately on each domain. (4) *Joint*: a base model is retrained from all the data of the domains seen so far in each incremental time step, breaking one of the constraints of the DIL. For a fair comparison, the base model on $\mathcal{D}_1$ is trained offline and on other domains trained online in incremental steps for all methods.

2.3. Online domain-incremental learning of devices using LoRA-BN combination

We propose to compute domain-specific LoRA parameters and BN statistics for ODIL. At the initial time step $t = 1$, the base model $\mathcal{M}$ is trained on dataset $\mathcal{D}_1$. At each incremental time step i, $\mathcal{M}$ is frozen and we only update its LoRA parameters and BN statistics using new dataset $\mathcal{D}_i$ as explained below.

Low-Rank Adaptation parameters

For a weight matrix $\boldsymbol{W}_{base} \in \mathbb{R}^{m \times n}$ of a convolutional layer of the base model $\mathcal{M}$, LoRA adds trainable rank decomposition matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ as:

$$W_{base} + \Delta W = W_{base} + AB, \tag{1}$$

where $A \in \mathbb{R}^{m \times r}$ is a down-projection matrix, $B \in \mathbb{R}^{n \times r}$ is a up-projection matrix and rank r is much smaller than the size of the inputs m and outputs n, i.e., $r \ll min(m, n)$. The forward pass of the network with LoRA changes from $\boldsymbol{W}_{base}\boldsymbol{x}$ to:

$$\boldsymbol{h} = \boldsymbol{W}_{base}\boldsymbol{x} + \boldsymbol{A}\boldsymbol{B}\boldsymbol{x}, \tag{2}$$

where $\boldsymbol{x}$ is the input and $\boldsymbol{h}$ is the hidden output. During incremental learning of a new domain, $\boldsymbol{W}_{base}$ is frozen and only the domain-specific weights of $\boldsymbol{A}$ and $\boldsymbol{B}$ are updated and stored in the model.

Statistics of Batch Normalization layer

BN normalizes the input activations of each layer using mini-batch statistics, i.e., running mean and variance. The behavior of the BN layer is different in the training and inference phases. During training, statistics of the BN layer are updated using training data forwarded through the network. During inference, statistics obtained from the training phase are fixed and used to standardize each layer of the network. BN performs well only when training and testing data come from the same domain. Therefore, we compute statistics for each domain separately and store into the model during training.

During inference at each incremental time step, domain-specific LoRA parameters and BN statistics are applied to base model $\mathcal{M}$ to classify acoustic scenes from the current domain, as shown in Fig. 1. Input to the model is a combination of the device ID and test sample, similar to task-incremental learning [12]. Device ID locates the LoRA parameters and BN statistics of the corresponding device before classifying the test sample. We only update additional LoRA parameters and statistics of the BN layers; all other parameters of the $\mathcal{M}$ are fixed. This allows us to recover the original performance of $\mathcal{M}$ for each device by replacing the corresponding LoRA parameters and BN statistics. Therefore, $\mathcal{M}$ does not suffer from forgetting previous devices when it learns a new device. Hereafter, we refer to our proposed approach as LoRA-BN.

3. EVALUATION AND RESULTS

3.1. Dataset and training setup

Experiments are conducted on the TAU Urban Acoustic Scenes 2020 Mobile development dataset [13], containing audio recordings from 3 real devices: denoted as A, B, and C, and additional S1-S6 devices simulated from device A. The domain $\mathcal{D}_1$ is composed of 40 hours of audio data from device A; the other 8 domains $\mathcal{D}_2$ to $\mathcal{D}_9$ include 3 hours of data each from devices B, C and S1-S6. We

Table 1: Device-specific accuracy of the different methods on each current domain.

Method	$\mathcal{D}_1$ A	$\mathcal{D}_2$ B	$\mathcal{D}_3$ C	$\mathcal{D}_4$ S1	$\mathcal{D}_5$ S2	$\mathcal{D}_6$ S3	$\mathcal{D}_7$ S4	$\mathcal{D}_8$ S5	$\mathcal{D}_9$ S6
Base	67.2	37.2	36.1	19.1	18.9	21.7	26.6	23.5	22.4
ODIL-BN [4]	67.2	40.8	44.7	23.1	22.5	26.0	30.5	31.2	25.4
LoRA-BN	67.2	**47.0**	**52.3**	**37.0**	**37.4**	**39.7**	**42.4**	**43.3**	**34.6**

Table 2: Average accuracy of the different methods over current and all previously seen domains.

Method	$\mathcal{D}_1$ A	$\mathcal{D}_2$ B	$\mathcal{D}_3$ C	$\mathcal{D}_4$ S1	$\mathcal{D}_5$ S2	$\mathcal{D}_6$ S3	$\mathcal{D}_7$ S4	$\mathcal{D}_8$ S5	$\mathcal{D}_9$ S6
FE	67.2	46.5	43.6	33.3	24.7	30.3	33.6	33.6	34.0
FT	67.2	48.0	48.4	37.7	33.1	39.0	43.9	43.4	44.0
Disjoint	67.2	48.0	46.6	35.3	29.1	35.9	36.2	34.9	36.9
LoRA-BN	67.2	**57.1**	**55.5**	**50.9**	**48.2**	**46.8**	**46.2**	**45.8**	**44.7**
Joint	67.2	60.3	59.7	56.3	56.1	55.0	56.4	56.7	54.7

follow the official training and testing split provided in the dataset to generate the data for each domain/device[1].

Initially, the model is trained on the domain $\mathcal{D}_1$ and it adapts to the remaining domains in incremental time steps. We follow the standard procedure in incremental learning, where the model is only trained on the current domain, without any data from previous domains, and evaluated on all previously seen domains.

3.2. Implementation details and evaluation metrics

We use the 6 convolutional blocks as a feature extractor and the layers specifications of each block are the same as PANNs CNN14 [14]. The global pooling is applied to the last convolutional layer to get a fixed-length input feature vector to the classifier. The entire network is trained from scratch on the first domain $\mathcal{D}_1$ as the base model. This base model is adapted to the other domains in incremental time steps. Input audio recordings are resampled to 32 kHz and log mel spectrograms are computed using default settings provided in [14].

The model is trained using the Adam optimizer [15] with a learning rate of 0.0001 and a mini-batch size of 32. The number of epochs to train the model on $\mathcal{D}_1$ is set to 120. The LoRA-BN and baselines are trained at incremental time steps for one epoch only. CosineAnnealingLR [15] scheduler updates the optimizer in every epoch. The rank r is set to 2 for minimal trainable parameters and the original kernel weight is 3.

We evaluate the performance of the model on the current domain and all previously seen domains at each incremental step using average accuracy and forgetting (Fr) as defined in [4]. Average accuracy is the average of accuracies of the method over the current and all previously seen domains. Average forgetting (Fr) is the average difference between the accuracy of the model for each domain at its learning iteration (the first time the model learns this domain) and the accuracy of the model for the same domain at the current iteration (after learning the current domain). A higher average accuracy and lower Fr are better.

[1]For S4-S6 the 3 hours of training data was not included in the official DCASE challenge train-test split, but is provided in the dataset.

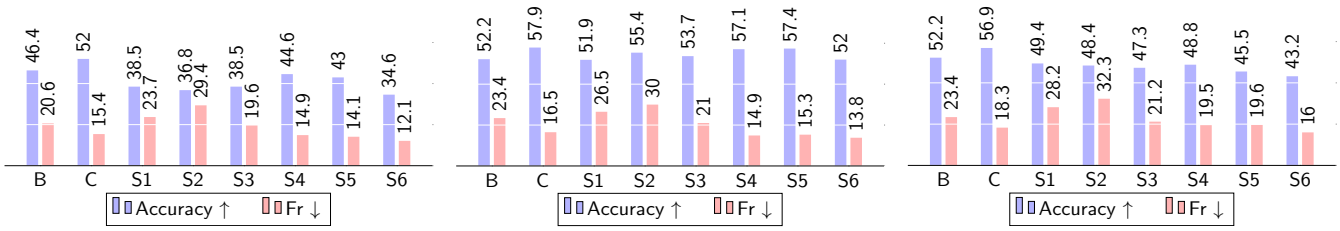
3.3. Results

The base model trained on data from real device A achieved an accuracy of 67.2% for domain $\mathcal{D}_1$. In Table 1, we compare the accuracy of proposed LoRA-BN on the current domain with other methods, in which all the parameters are frozen or only a few device-specific parameters are updated in incremental steps and therefore not suffer from forgetting.

To check the severity of the mismatch between domain $\mathcal{D}_1$ and other incremental domains $\mathcal{D}_2$ to $\mathcal{D}_9$, we use the base model to classify the acoustic scenes of other domains without updating its parameters (no training). The base model does not adapt to the incremental domains, resulting in a drastic performance drop, especially from simulated domains, $\mathcal{D}_4$ to $\mathcal{D}_9$, as seen in Table 1.

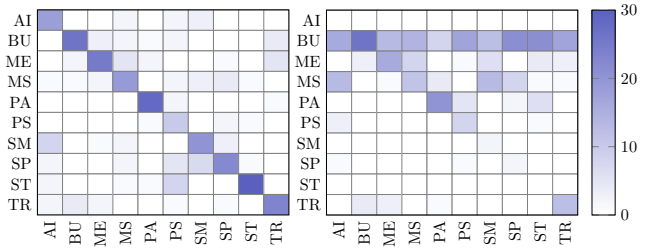
ODIL-BN computes the domain-specific running statistics of the BN layers to classify acoustic scenes from each domain [4]. ODIL-BN does not change any other parameters of the base model and does not forget previous domains. However, this alone improves the performance of the base model only slightly in most of the incremental domains. The proposed LoRA-BN computes the domain-specific LoRA parameters for each convolutional layer and domain-specific running statics for each BN layer. The additional combined LoRA parameters and running statistics help the base model to effectively adapt to the incremental domains. It can be seen that LoRA-BN improves the performance for $\mathcal{D}_2$ by 9.8%p (percentage point), $\mathcal{D}_3$ by 16.2%p, $\mathcal{D}_4$ by 17.9%p, $\mathcal{D}_5$ by 18.5%p, $\mathcal{D}_6$ by 18.0%p, $\mathcal{D}_7$ by 15.8%p, $\mathcal{D}_8$ by 19.8%p, $\mathcal{D}_9$ by 12.2%p, compared to the base model.

We also compare the performance of the proposed LoRA-BN method with other popular baseline methods in terms of average accuracy over current and previous domains in Table 2. Accuracy in the current domain and average forgetting over previous domains is also shown in Fig. 2. Results of FE compared to FT show that adapting the layers of the feature extractor to an incremental domain is better than freezing them. One can observe from Fig. 2a and 2b that higher forgetting of previous real domains $\mathcal{D}_1$ to $\mathcal{D}_3$ happens when the model starts learning the simulated domains, specifically $\mathcal{D}_4$ and $\mathcal{D}_5$ due to highly mismatched domains. The poor performance of FT in classifying the acoustic scenes from $\mathcal{D}_1$ after learning $\mathcal{D}_5$ can also be seen in Fig. 3b. This leads to a lower average accuracy for



(a) FE (b) FT (c) Disjoint

Figure 2: Accuracy at the current domain and average forgetting over previous domains of FE (a), FT (b) and disjoint (c) methods.



(a) (b)

Figure 3: Confusion matrices of a base model on domain $\mathcal{D}_1$ (a), FT on domain $\mathcal{D}_1$ after learning the simulated domain $D5$ corresponding to S2 (b). The 10 classes are, AI: airport, BU: bus, ME: metro, MS: metro station, PA: park, PS: public square, SM: shopping mall, SP: pedestrian street, ST: street with traffic, and TR: tram.

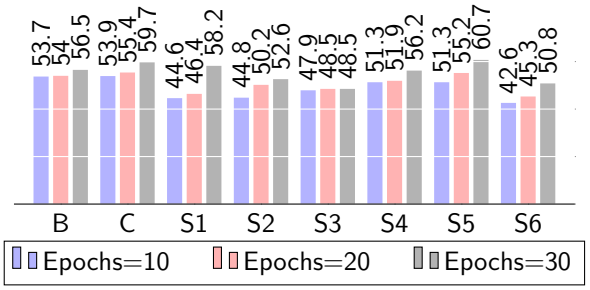


Figure 4: Accuracy of the LoRA-BN over increasing number of epochs.

FE and FT for simulated domains $\mathcal{D}_4$ and $\mathcal{D}_5$, as seen in Table 2. However, FT uses all layers to adapt to the simulated devices after $\mathcal{D}_5$,and performs better overall after learning all domains.

The *disjoint* approach fine-tunes the base model trained on $\mathcal{D}_1$ to a current domain and performs well on real domains $\mathcal{D}_2$ and $\mathcal{D}_3$, maybe due to similar feature distributions. However, fine-tuning the base model directly to each simulated domain reduces the performance of the disjoint method as compared to FT, in which previous knowledge of the simulated domain is used to classify the acoustic scenes from the current domain.

The proposed LoRA-BN outperforms all other methods without forgetting any of the previously learned domains and its performance is close to the baseline joint which trains the model from the data of all previously seen domains. The number of LoRA parameters for each domain is 124434, which is only a 0.17% increase to the total parameters 75497930 of the base model. It shows that LoRA-BN is more suitable for practical scenarios because it only stores inexpensive LoRA parameters and running statistics.

We also compare the performance of LoRA-BN and other baseline systems in offline settings. Baseline systems suffer from overfitting and lead to decreased performance. However, LoRA-BN converges effectively over an increasing number of epochs with limited training data in incremental domains, as seen in Fig. 4. Further, we test the performance of all the methods by changing the order of the domains. We found that the devices S1 and S2 are more challenging to adapt in any order than other devices.

In comparison to the results of the DCASE Challenge 2020[2], the baseline achieves an average accuracy of 54.1%, being trained for 200 epochs on combined data of devices A-S3, with S4-S6 not included in the training. This result is aligned with the joint baseline in this paper, which achieves 54.7% using online training of all devices using the base model. DCASE baseline reports lower performance on simulated devices S1-S3, being trained offline, non-incrementally. Our proposed LoRA-BN achieves comparable results on S1-S3 when trained for 30 epochs, only on data of one device sequentially. However, our method follows a completely different learning procedure and is therefore not fully comparable with the DCASE baseline.

[2] https://dcase.community/challenge2020/task-acoustic-scene-classification#subtask-a-3

4. CONCLUSION

In this paper, we propose a combination of LoRA parameters and running statistics of the BN layer for ODIL of acoustic scenes from different devices over time. Results show that highly mismatched simulated devices, especially starting devices S1 and S2 are more difficult to adapt by a model trained on real devices. ODIL-BN achieves poor performance on simulated devices and baselines severely forget acoustic scenes from previous real devices when these start learning simulated devices. The proposed LoRA-BN adapts effectively to the new domain and increases the performance of the base model by a large margin without forgetting acoustic scenes from any of the previously leaned devices. The performance of the LoRA-BN is further improved by increasing the number of iterations over the training data. LoRA-BN stores and uses inexpensive parameters and is more suitable for realistic applications. Future works include the development of a domain-agnostic approach that does not require device ID to classify acoustic scenes.

5. REFERENCES

[1] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural networks*, vol. 113, pp. 54–71, 2019.

[2] M. J. Mirza, M. Masana, H. Possegger, and H. Bischof, "An efficient domain-incremental learning approach to drive in all weather conditions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3001–3011.

[3] P. Garg, R. Saluja, V. N. Balasubramanian, C. Arora, A. Subramanian, and C. Jawahar, "Multi-domain incremental learning for semantic segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 761–771.

[4] M. Mulimani and A. Mesaros, "Online domain-incremental learning approach to classify acoustic scenes in all locations," in *European Signal Processing Conference (EUSIPCO)*. IEEE, 2024.

[5] S. Gharib, K. Drossos, E. Cakir, D. Serdyuk, and T. Virtanen, "Unsupervised adversarial domain adaptation for acoustic scene classification," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018, pp. 138–142.

[6] K. Drossos, P. Magron, and T. Virtanen, "Unsupervised adversarial domain adaptation based on the wasserstein distance for acoustic scene classification," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 259–263.

[7] A. I. Mezza, E. A. Habets, M. Müller, and A. Sarti, "Unsupervised domain adaptation for acoustic scene classification using band-wise statistics matching," in *28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 11–15.

[8] E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 1–9.

[9] M. Wistuba, L. Balles, G. Zappella, *et al.*, "Continual learning with low rank adaptation," in *NeurIPS Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.

[10] S. Aleem, J. Dietlmeier, E. Arazo, and S. Little, "Convlora and adabn based domain adaptation via self-training," *arXiv preprint arXiv:2402.04964*, 2024.

[11] Z. Zhong, Z. Tang, T. He, H. Fang, and C. Yuan, "Convolution meets lora: Parameter efficient finetuning for segment anything model," in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[12] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[13] T. Heittola, A. Mesaros, and T. Virtanen, "Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020, pp. 56–60.

[14] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[15] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations (ICLR)*, 2017.