

# Domain Information Control at Inference Time for Acoustic Scene Classification

<sup>1,2</sup>Shahed Masoudian, <sup>2</sup>Khaled Koutini, <sup>1,2</sup>Markus Schedl, <sup>1,2</sup>Gerhard Widmer, <sup>1,2</sup>Navid Rekasaz

<sup>1</sup>Johannes Kepler University Linz, Institute of Computational Perception

<sup>2</sup>Linz Institute of Technology (LIT), AI Lab  
Austria

**Abstract**—Domain shift is considered a challenge in machine learning as it causes significant degradation of model performance. In the Acoustic Scene Classification task (ASC), domain shift is mainly caused by different recording devices. Several studies have already targeted domain generalization to improve the performance of ASC models on unseen domains, such as new devices. Recently, the Controllable Gate Adapter (CONGATER) has been proposed in Natural Language Processing to address the biased training data problem. CONGATER allows controlling the debiasing process at inference time. CONGATER’s main advantage is the *continuous* and *selective* debiasing of a trained model, during inference. In this work, we adapt CONGATER to the audio spectrogram transformer for an acoustic scene classification task. We show that CONGATER can be used to selectively adapt the learned representations to be invariant to device domain shifts such as recording devices. Our analysis shows that CONGATER can progressively remove device information from the learned representations and improve the model generalization, especially under domain shift conditions (e.g. unseen devices). We show that information removal can be extended to both device and location domain. Finally, we demonstrate CONGATER’s ability to enhance specific device performance without further training<sup>1</sup>.

**Index Terms**—Acoustic Scene Classification, Domain Adaptation, Transformers, Adapters, ConGater

## I. INTRODUCTION

Domain Generalization is a critical topic in Deep Neural Networks (DNN). The performance of conventional DNN methods drastically degrades under domain shift conditions when evaluated on new domains [1]. Therefore, the ability of machine learning models to generalize to these new and unseen domains is crucial in real-world applications. Methods to improve the generalization of DNNs to a new domain are well studied in different fields, such as computer vision [2], audio perception [3] and natural language processing [4].

This problem of domain generalization has drawn the attention of the Detection and Classification of Audio Events (DCASE) community, and datasets were constructed to test the generalization of common machine learning models under domain shift conditions [5].

Convolutional Neural Networks (CNNs) have dominated the acoustic scene classification literature [3], [6]. More recently, vision transformers were adapted to audio tasks and shown

to outperform CNNs on ASC tasks [7]. Schmid et al. [8] fine-tuned a transformer model pre-trained on a large dataset on an ASC dataset and used it as a teacher for knowledge distillation to low-complexity CNNs. Kim et al. [9] adapted a CNN model to this task. In order to address the domain generalization problem, Schmid et al. [8] used Frequency Mixstyle data augmentation to improve the performance on unseen devices. Kim et al. [3] introduced ResNorm in BC-ResNet and Relaxed Instance Frequency Normalization (RFIN) as a new normalization method to achieve state-of-the-art results on unseen devices. Another approach to tackle the Domain Generalization issue is unsupervised Domain Adaptation (DA). Using Wasserstein distance, Drossos et al. [10] learned domain invariant feature representations and improved the result of the ASC model on an unseen domain. Gharib et al. [11] also used adversarial training to learn domain invariant representations to improve the performance of the ASC model on unseen devices.

Invariant representation learning is also widely studied in the field of Natural Language Processing (NLP) on scenarios such as domain adaptation [4], domain transfer [12], and mitigation of societal biases. In particular, the bias mitigation methods aim at removing sensitive information (e.g. Gender) from the DNN embeddings to improve fairness. The introduced approaches share many conceptual and methodological commonalities with domain invariant representation in DA, such as the utilization of adversarial techniques [13]. These approaches are widely studied in NLP literature [14]–[16]. Recent studies in this field focus on modular neural networks, where end-users can choose between debiased and biased models. These methods approach this by adding a separate module such as Adapters [17], [18], or sparse subnetworks [19], [20] to network. In this case, instead of training the whole model, these separate modules are trained to improve training efficiency and add a modular capability to the network [18]. More recently, Controllable Gate Adapter [21] (CONGATER) expand the mentioned work by providing the ability for continuous sensitive information removal from the trained model.

In this work, we adopt the CONGATER idea from NLP and apply it to audio tasks. By adding CONGATER in between audio spectrogram transformer layers. We aim to remove device and location information to improve the generalization of audio spectrogram transformers on ASC tasks under domain shift conditions. We show that (1) CONGATER can effectively

This work received financial support by the State of Upper Austria and the Federal Ministry of Education, Science, and Research through grant LIT-2021-YOU-215 and basic funding of the LIT AI Lab.

<sup>1</sup>Source Code: [https://github.com/ShawMask/dcasc22\\_congater](https://github.com/ShawMask/dcasc22_congater)

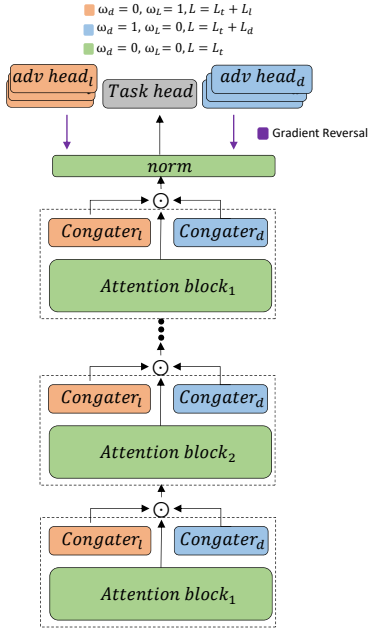


Fig. 1: Our proposed model using CONGATER modules, the location of the CONGATER layers, and the trainable parameters in each training step. The green blocks are trained with the loss of task ( $L_t$ ), the blue blocks are trained adversarially to remove device information using task loss and reversed gradient loss of the device ( $L_d$ ), the orange blocks are also trained adversarially to remove location information using task and reversed gradient location loss ( $L_l$ ). For each domain, the average loss of the three adversarial heads is considered as domain loss.

control the amount of information of device and location in the network on a continuous range; (2) by removing information from the device, we achieve on average 0.7% higher overall accuracy compared to the baseline model, and 1.1% accuracy improvement on unseen devices indicating the better generalization of the network; (3) our method can simultaneously be applied to both device and location domains; additionally, we demonstrate CONGATER’s ability to fine-tune on a specific device by selecting suitable hyper-parameter at inference time without any need for further training.

## II. METHOD

For our experiments, we choose a transformer-based model PASST [7] that performs very well on a wide range of audio tasks, including ASC tasks [22]. PASST have a similar structure to Vision Transformers [23]. PASST works by extracting patches of an input spectrogram and linearly projecting them to a higher dimension corresponding to the embedding size. The positional encoding—consisting of time and frequency positional encodings—are added to these embeddings. Patchout [7] is then applied to speed up the training and for regularization. The sequence is then passed through several self-attention layers and a feed-forward classifier. In our experiments, we use the pre-trained model on Audioset [24] and then fine-tune the model on the ASC task. Since the

CONGATER [21] was introduced as modules added to the transformer encoder layers of BERT [25] language model, we similarly extend PASST with CONGATER by adding the CONGATER modules after the attention blocks of PASST, as explained in the following.

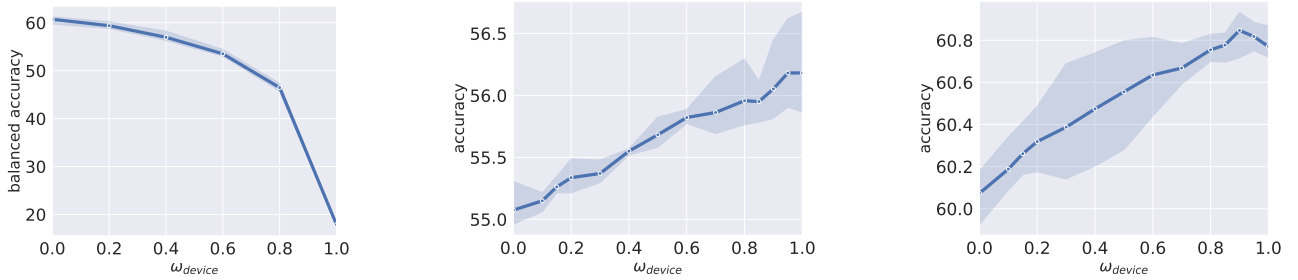
CONGATER [21] extends the core idea of Adapter networks [17] by introducing a novel controllable gating mechanism, applied to input embeddings to deliver the desired learning objective such as learning a task or mitigating bias. Each CONGATER layer consists of a feed-forward network followed by a novel activation function called *trajectory-sigmoid* (t-sigmoid). t-sigmoid is similar to normal sigmoid function, but with the extra sensitivity parameter  $\omega$ , as formulated below:

$$\text{t-sigmoid}(v) = 1 - \frac{\log_2(\omega + 1)}{1 + e^v} \quad \omega \in [0, 1] \quad (1)$$

The parameter  $\omega$  can be manipulated manually at inference time and smoothly change the shape of the activation function. At  $\omega = 0$ , the shape of t-sigmoid is the constant  $y = 1$  function. As  $\omega$  increases, the activation function smoothly reshapes to a sigmoid function, adding a stronger non-linearity to the input.

In order to remove information of several domains, a CONGATER is dedicated to each domain and added to the network. As an instance, for the domain *location*, the gating vector  $\mathbf{g}_{location}$  is achieved after applying the feed forward layer followed by t-sigmoid. The overall gate output is the element-wise multiplication of the vectors, which in our case is defined as  $\mathbf{g} = \mathbf{g}_{device} \odot \mathbf{g}_{location}$ . The final output the layer is defined as the self-gate between the original input to CONGATER (namely the output of the attention layer denoted by  $\mathbf{x}$ ), and the final gating vector:  $output = \mathbf{x} \odot \mathbf{g}$ . Note that self-gate between any input and CONGATER of a domain with  $\omega_d = 0$  do not affect the input. By increasing  $\omega_d$ , the effect of removing this specific domain – independent of other domain(s) – increases. In our transformer-based architecture as shown in Figure 1, we add one CONGATER layer after each of the attention blocks.

The training sequence for the proposed model consists of three steps, executed sequentially as each batch of training data arrives. The first step is **Task Training**, where parameters of the PASST in addition to the task-head are trained with  $\omega_{device} = 0$  and  $\omega_{location} = 0$ . The second step is **Device Removal**, in which the parameters of the device CONGATER in addition to the task-head are trained with  $\omega_{device} = 1$  and  $\omega_{location} = 0$  to learn the task and remove the device-related information. Finally, in the third step **Location Removal**, the parameters of the location CONGATER in addition to the ones of the task-head are trained with  $\omega_{device} = 0$  and  $\omega_{location} = 1$  in order to learn the task and remove location-related information. We utilized Domain Adversarial Neural Network (DANN) [13] to remove information from embeddings. In this method, an additional sub-network (*adversarial head*) is added to predict device/location labels from the original network embeddings. The reversed gradient loss is added to the network as an extra objective ( $L_{total} = L_{task} + L_{adv}$ ). This objective—because of the gradient reversal—aims at removing



(a) Device probe balanced accuracy

(b) Task accuracy on unseen devices

(c) Task accuracy on all devices

Fig. 2: (a) Balanced accuracy of the device probe as we increase  $\omega_{device}$  (b) model accuracy on unseen devices (S4-S6) as we increase only the  $\omega_{device}$ . (c) Overall accuracy of the model on validation set as we increase only  $\omega_{device}$

the domain information by maximizing the domain confusion in the original network embedding.

### III. EXPERIMENT SETUP

**Dataset.** The dataset used for our experiments is the adapted version of TAU Urban Acoustic Scenes 2022 Mobile development dataset used for the Acoustic Scene Classification (ASC) task [5] in the DCASE 2022 challenge. The dataset contains 1-second audio recordings of 10 different acoustic scenes. The development set consists of audio recordings from 9 different devices, 3 real devices (A, B, C), and 6 simulated devices (S1-S6). The audios are recorded in 12 different European cities. We split 70% of the development set for training in which only 6 devices (A, B, C, S1-S3) are used for training and the other 3 (S4-S6) are only seen during validation, which we refer to as *unseen devices*. Similarly for the cities, during training 10 cities are available and 2 cities are available only during validation.

**Preprocessing.** We use a sampling rate of 32kHz. We apply Short Time Fourier Transformation (STFT) with a window size of 800 with an overlap of 320 (40%) to generate the spectrograms. We apply mel-filters bank 128 in a similar setup to [8]. The final spectrograms have 128 mel-frequency bins and 100 time frames.

**Training.** We train each model with a batch size of 100 with *CrossEntropy* loss, *Adamw* optimizer, weight decay of 0.001, max learning rate of  $1 \times 10^{-5}$  for PASST model, and  $1 \times 10^{-4}$  for the CONGATER layers. We use a pre-trained PASST model on Audioset [24]. We initialize all CONGATER layers with zero weight and bias  $b = 5$ . This initialization forces the output of CONGATER layers to 1 at the beginning of training. Each model is trained for 25 epochs with the following learning rate scheduler: for the first 3 epochs, the learning rate is exponentially increasing to its max value, in the next 3 epochs the model is trained with the max learning rate, and finally for the next 10 consecutive epochs the learning rate decreases to reach 0.01 of the max value. Adversarial training is used for CONGATER layers to remove device/location information. In order to have a smooth adversarial gradient flow during training, we use average the loss of three adversarial heads. No normalization, augmentation, or dataset balancing is applied, in order to be able to check the direct effect of information removal on the baseline model where  $\omega_{device} = 0$  and  $\omega_{location} = 0$ .

In this section, we discuss and analyze the results of our experiments. As mentioned in the previous sections, we train our proposed model, and the two independent CONGATER modules are trained to remove the domain information of the device and location. More concretely, at inference time each CONGATER parameter  $\omega$  is used to remove the corresponding domain information from the embeddings of the network by increasing its value from zero to one. The parameters  $\omega_{device}$ ,  $\omega_{location}$  control the intensity of the domain adaptation process for the device and the location, respectively. In our experiments, we mainly focus on the effect of  $\omega_{device}$  on the network, namely in terms of device information leakage, and model performance evaluated on all and unseen devices. In addition, we report and analyze the results when controlling simultaneously over both device and location domains. To account for the possible variations in the results, we repeat each experiment three times and report the average and standard deviation of the evaluation results.

#### A. Device Information

In order to check whether domain information is indeed removed from the network, we evaluate the degree of information leakage using additional classifier heads. This method is commonly referred to as *probing* in the NLP literature [15]. Each probe is a two-layer feed-forward network with a ReLU activation function and is trained on the output embedding of the last attention layer of the network, before the task head. The probe aims to classify device labels from the embeddings. We retrain the probe for each value of  $\omega_{device}$ . The probes are trained for 5 epochs with a learning rate of  $1 \times 10^{-4}$ . We calculate the balanced accuracy of the device classification on the validation set and report the average and standard deviation of the results attained from three independent runs.

Figure 2a shows that at  $\omega_{device} = 0$ , namely the Baseline with no information removal from the model, the average balanced accuracy of the probe for device label is 60.7%. This indicates that the embeddings of the model are highly informative about the recording device of the incoming audio. As we increase the  $\omega_{device}$  and retrain the probes, the balanced accuracy consistently drops, indicating that the device information is decreasing in the embeddings. At  $\omega_{device} = 1$ , we reach the lowest balanced accuracy of 17.9%, showing that the CONGATER module has successfully removed most of the

Model	Unseen Devices			Seen Devices					Unseen	Overall		
	S4	S5	S6	A	B	C	S1	S2			S3	
Baseline	56.9 <sub>0.6</sub>	57.3 <sub>0.4</sub>	50.9 <sub>1.3</sub>	72.6 <sub>0.3</sub>	63.5 <sub>0.5</sub>	67.5 <sub>0.4</sub>	58.0 <sub>0.9</sub>	55.6 <sub>0.8</sub>	58.2 <sub>0.5</sub>	55.1 <sub>0.2</sub>	60.1 <sub>0.1</sub>	
$\omega_d = 0.6, \omega_l = 0.0$	57.4 <sub>0.5</sub>	<b>58.1</b> <sub>0.4</sub>	51.9 <sub>0.8</sub>	72.7 <sub>1.2</sub>	63.5 <sub>0.4</sub>	67.7 <sub>0.3</sub>	59.4 <sub>0.7</sub>	56.2 <sub>0.6</sub>	58.7 <sub>0.3</sub>	55.8 <sub>0.1</sub>	60.6 <sub>0.2</sub>	
$\omega_d = 0.7, \omega_l = 0.0$	57.3 <sub>0.4</sub>	58.0 <sub>0.2</sub>	52.2 <sub>1.0</sub>	72.6 <sub>1.1</sub>	63.5 <sub>0.6</sub>	<b>67.8</b> <sub>0.3</sub>	59.5 <sub>0.5</sub>	56.2 <sub>0.6</sub>	58.8 <sub>0.3</sub>	55.9 <sub>0.3</sub>	60.7 <sub>0.1</sub>	
$\omega_d = 0.9, \omega_l = 0.0$	57.5 <sub>0.3</sub>	57.9 <sub>0.4</sub>	52.7 <sub>0.9</sub>	72.8 <sub>0.7</sub>	63.6 <sub>0.2</sub>	67.5 <sub>0.4</sub>	<b>59.7</b> <sub>0.6</sub>	<b>56.6</b> <sub>0.7</sub>	59.2 <sub>0.6</sub>	56.1 <sub>0.3</sub>	<b>60.8</b> <sub>0.1</sub>	
$\omega_d = 0.9, \omega_l = 0.1$	57.4 <sub>0.2</sub>	58.1 <sub>0.5</sub>	52.7 <sub>1.0</sub>	72.7 <sub>0.7</sub>	<b>63.8</b> <sub>0.6</sub>	67.6 <sub>0.4</sub>	59.5 <sub>1.0</sub>	56.3 <sub>0.4</sub>	59.0 <sub>0.4</sub>	56.1 <sub>0.5</sub>	60.8 <sub>0.3</sub>	
$\omega_d = 1.0, \omega_l = 0.1$	<b>57.7</b> <sub>0.3</sub>	58.1 <sub>0.7</sub>	<b>53.0</b> <sub>1.0</sub>	<b>72.8</b> <sub>0.3</sub>	63.7 <sub>0.7</sub>	67.3 <sub>0.5</sub>	59.4 <sub>0.9</sub>	56.1 <sub>0.4</sub>	<b>59.3</b> <sub>0.6</sub>	<b>56.3</b> <sub>0.6</sub>	60.8 <sub>0.3</sub>	
Device-specific Tuning achieved in $[\omega_d, \omega_l]$	<b>57.9</b> <sub>0.3</sub> [1.0,0.1]	<b>58.1</b> <sub>0.4</sub> [0.6,0.]	<b>53.0</b> <sub>1.0</sub> [1.0,0.1]	<b>72.8</b> <sub>0.3</sub> [1.0,0.1]	<b>63.8</b> <sub>0.6</sub> [0.9,0.1]	<b>67.8</b> <sub>0.3</sub> [0.7,0.]	<b>59.7</b> <sub>0.6</sub> [0.9,0.0]	<b>56.6</b> <sub>0.7</sub> [0.9,0.0]	<b>59.3</b> <sub>0.6</sub> [1.0,0.1]	<b>56.3</b> <sub>0.6</sub> [1.0,0.1]	-	-

TABLE I: Comparison of the performance of our CONGATER-based model on target devices under various values of the  $\omega$  parameters. We group devices as **Seen**, and **Unseen**, indicating whether they exist during training or only appear in evaluation. The subscripts  $d$  and  $l$  refer to device and location, respectively. Baseline is the case with  $\omega_d = \omega_l = 0.0$ , and Device-specific Tuning refers to dynamically find the best performing  $\omega$  combination for a specific target device.

information about the device, as – in contrast to the Baseline – the probe is not able to predict device labels with high accuracy. Our results show that CONGATER can effectively remove domain information from the model, and the amount of information removal can be decided at inference time over a continuous range.

### B. Model Performance

The main purpose of domain adaptation in ASC is to achieve better performance on unseen domains by generating domain invariant feature representation. We now turn our attention to the task accuracy of the model on unseen devices (S4-S6). Figure 2b shows the task accuracy results of the network on unseen devices as we increase  $\omega_{device}$ . We observe that from the Baseline position  $\omega_{device} = 0$  (where no information from the device is removed), the average task accuracy on unseen devices starts at 55.1%. By increasing  $\omega_{device}$  and removing the device information, the accuracy of the network on unseen devices continuously improves and reaches its max at 56.2% when  $\omega = 1$ . The results on unseen accuracy indicate that increasing  $\omega$  and removing more device information leads to improving model generalization across unseen devices. We also evaluate the overall (all devices) performance of the model with the same procedure as before. Looking at Figure 2c, we see that by increasing  $\omega_{device}$ , task accuracy increases and peaks at  $\omega_{device} = 0.9$  where the average task accuracy is 60.8%, followed by a negligible drop at  $\omega_{device} = 1$ . All in all the results are a clear indication that by removing device information from the network we achieve less informative embeddings which leads to better generalization on unseen devices as well as overall model performance.<sup>2</sup>

### C. Device Specific Tuning

In this section, we discuss the benefits of controllable information removal at inference, particularly in the scenario of finding the optimal model configuration according to a specific target domain. As shown in Figure 3 and similar to previous experiments (Figure 2a), CONGATER-based model can adjust the information of both domains without additional training. Interestingly, the simultaneous removal of both factors on the top-right of the plot ( $\omega_{device} = 1$  and  $\omega_{location} = 1$ ) leads to a

<sup>2</sup>Extensive 2D analysis of the domains is available in the GitHub repository

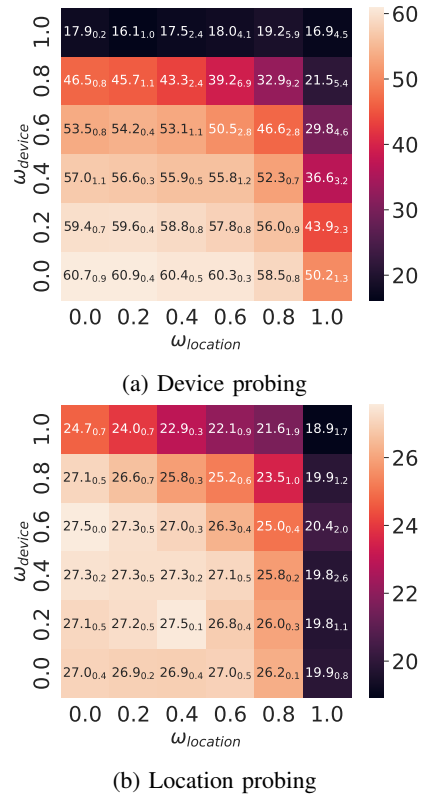


Fig. 3: Balanced accuracy of the probes, measuring the information leakage of device (a) and location (b) in our model, when increasing device and location  $\omega$  parameters.

less information leakage of both device and location domains than information removal of one domain.

Despite the successful removal of domain-specific information, our evaluation of task accuracy shows that removing only the location (in contrast to the device) does not result in meaningful performance improvement. However, CONGATER’s ability to manipulate domain information can be leveraged to enhance model performance for a specific domain. We observed that by careful selection of the  $\omega_{device}$  and  $\omega_{location}$ , we can improve model’s performance with regard to one or several devices. This ability can be particularly advantageous when we introduce a new unknown device with a small subset of audio. Instead of re-training the whole model for a new device, we can simply select the suitable  $\omega$  for

the same pre-trained model at inference without any need for further training. The selection process can be either based on expert knowledge of the source and the target domain or by empirically validating model performance on a small subset of the target domain at inference (Grid search of the CONGATER parameter  $\omega$ ). Table I shows the specific values of  $\omega_{device}$ ,  $\omega_{location}$  and model performance on each device. As it can be seen from the table there are unique  $\omega_{device}$  and  $\omega_{location}$  which result in better performance of the same model for one particular device. For instance, even though the best  $\omega$  value for unseen devices is  $\omega_{device} = 1$ ,  $\omega_{location} = 0.1$ , we can see that for device S2, model has 0.5% better performance on average at  $\omega_{device} = 0.9$ ,  $\omega_{location} = 0.0$ . Note that with device-specific tuning, it is possible to optimize for a specific device, albeit at the expense of the performance of other devices.

#### IV. CONCLUSION

In this paper, we adopt CONGATER from the NLP domain and implement it on audio spectrogram transformers. Training CONGATER on ASC dataset for multi-device domain adaptation, we observe that our model using CONGATER modules is effective at continuous device/location information removal from the network embedding at inference time. We observe that by increasing the sensitivity of the parameter  $\omega_{device}$ , the embeddings of the network effectively lose device information (domain labels), shown by the decrease in the balanced accuracy of the probes. We observe significant improvement in both unseen devices and overall accuracy for an already trained model by adjusting  $\omega_{device}$  from 0 to 1. This observation indicates that removing device information leads to a more robust embedding for unseen devices. We observe that removing information from location alone does not significantly improve task accuracy, nor the unseen accuracy of the devices. Finally, we demonstrate that correct selection of CONGATER hyperparameter for location and device leads to device-specific performance improvement.

#### REFERENCES

- [1] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.
- [2] H. Venkateswara and S. Panchanathan, *Domain adaptation in computer vision with deep learning*. Springer, 2020.
- [3] B. Kim, S. Yang, J. Kim, H. Park, J. Lee, and S. Chang, "Domain generalization with relaxed instance frequency-wise normalization for multi-device acoustic scene classification," in *Interspeech*, 2022.
- [4] D. Jin, Z. Jin, Z. Hu, O. Vechtomova, and R. Mihalcea, "Deep learning for text style transfer: A survey," *Computational Linguistics*, vol. 48, no. 1, pp. 155–205, 2022.
- [5] I. Martín-Morató, F. Paissan, A. Ancilotto, T. Heittola, A. Mesaros, E. Farella, A. Brutti, and T. Virtanen, "Low-complexity acoustic scene classification in DCASE 2022 challenge," in *Proceedings of the 7th Workshop on Detection and Classification of Acoustic Scenes and Events 2022, DCASE 2022, Nancy, France, November 3-4, 2022*, 2022.
- [6] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "CP-JKU submissions to DCASE'20: Low-complexity cross-device acoustic scene classification with rf-regularized cnns," *Tech. Rep., DCASE2020 Challenge*, 2020.
- [7] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *Interspeech*, 2022.
- [8] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.
- [9] J.-H. Lee, J.-H. Choi, P. M. Byun, and J.-H. Chang, "Multi-scale architecture and device-aware data-random-drop based fine-tuning method for acoustic scene classification," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.
- [10] K. Drossos, P. Magron, and T. Virtanen, "Unsupervised adversarial domain adaptation based on the wasserstein distance for acoustic scene classification," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.
- [11] S. Gharib, K. Drossos, E. Cakir, D. Serdyuk, and T. Virtanen, "Unsupervised adversarial domain adaptation for acoustic scene classification," in *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events, DCASE 2018, Surrey, UK, November 19-20, 2018*, M. D. Plumley, C. Kroos, J. P. Bello, G. Richard, D. P. W. Ellis, and A. Mesaros, Eds., 2018.
- [12] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, "Transfer learning in natural language processing," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 15–18.
- [13] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [14] D. Kashyap, S. K. Aithal, C. Rakshith, and N. Subramanyam, "Towards domain adversarial methods to mitigate texture bias," in *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*.
- [15] D. Kumar, O. Lesota, G. Zerveas, D. Cohen, C. Eickhoff, M. Schedl, and N. Rekasaz, "Parameter-efficient modularised bias mitigation via adapterfusion," in *Proceeding of 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2023.
- [16] H. Yuan, J. Zheng, Q. Ye, Y. Qian, and Y. Zhang, "Improving fake news detection with domain-adversarial and graph-attention neural network," *Decision Support Systems*, vol. 151, p. 113633, 2021.
- [17] N. Houslyby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *International Conference on Machine Learning*, vol. 97. Proceedings of Machine Learning Research, 09–15 Jun 2019.
- [18] A. Lauscher, T. Lueken, and G. Glavaš, "Sustainable modular debiasing of language models," in *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021.
- [19] L. Hauenberger and N. Rekasaz, "Parameter efficient diff pruning for bias mitigation," *arXiv preprint arXiv:2205.15171*, 2022.
- [20] J. M. Meissner, S. Sugawara, and A. Aizawa, "Debiasing masks: A new framework for shortcut mitigation in NLU," in *Proceeding of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.
- [21] S. Masoudian, O. Lesota, M. Schedl, and N. Rekasaz, "Controllable attribute removal for continuous bias mitigation at inference time," 2023, anonymous preprint under review. [Online]. Available: <https://openreview.net/pdf?id=aQ1gwltlNP>
- [22] F. Schmid, K. Koutini, and G. Widmer, "Efficient Large-scale Audio Tagging via Transformer-to-CNN Knowledge Distillation," *arXiv preprint arXiv:2211.04772*, 2022.
- [23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [24] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.
- [25] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019.