Unsupervised Feature-Space Domain Adaptation applied for Audio Classification

Amir Latifi Bidarouni
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany
amir.latifi.bidarouni@idmt.fraunhofer.de

Jakob Abeßer
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany
jakob.abesser@idmt.fraunhofer.de

***Abstract*—Domain adaptation is a fundamental technique to ensure that deep neural networks perform robustly even in unknown target domains. In this paper, we study Z-Score normalization, relaxed instance frequency-wise normalization (RFN), and feature projection-based DA (FPDA) for unsupervised feature-based domain adaptation. With a focus on acoustic monitoring, we investigate the classification of individual sounds and acoustic scenes as the main use cases. Based on a systematic study of different normalization techniques and data partitioning strategies, our results confirm that an individual normalization per frequency band is beneficial for sound classification, whereas a global classification applied to individual data instances is beneficial for acoustic scene classification. As another main contribution, we propose the IFPDA method, essentially, is a variation of the original FPDA configuration, allowing it to be applied independently to each instance, and results in a substantial performance improvement and even outperforms all other normalization methods in the acoustic scene classification task.**

***Index Terms*—Domain adaptation, data normalization, sound classification, acoustic scene classification**

I. Introduction

Deep learning techniques have gained wide acceptance in the analysis of audio, image, video, and textual data since they have been shown to consistently outperform traditional methods. In an Internet-of-Sounds (IoS) context, autonomous acoustic sensors with different characteristics are often deployed at different recording locations. A key challenge in tasks such as Sound Event Detection (SED) and Acoustic Scene Classification (ASC) is to ensure that classification models are robust towards out-of-distribution data. A common assumption in machine learning is that training and test data are drawn from the same distribution. However, in many real-world scenarios, this assumption does not hold. Generally, one distinguishes between the source domain (SD) data, i. e., the labeled data set with which the model is trained, and the target domain (TD) data, i. e., the data with which the model is evaluated. In a microphone mismatch scenario, where audio data is recorded with different recording devices, the corresponding distributions do not match and the performance of the model is impaired due to the resulting "domain shift".

One possible solution to mitigate the domain shift problem is to carefully select and/or modify the SD data such that it better represents the targeted use case (TD). This can be achieved, for example, by generating variations of existing training examples (data augmentation). As a second solution, domain adaptation (DA) techniques can be applied to improve the generalization capabilities of a trained model toward unseen data. In supervised DA, the labeled data from both the source and target domain are used to adjust the model parameters in order to learn domain-invariant feature representations. In unsupervised DA, only data, but no annotations from the TD are available.

Furthermore, model-based and feature-based DA methods can be distinguished. In model-based DA, the generalization of a model can be improved, for instance, by modifying parts of its architecture or by imposing regularization constraints during training. As a main disadvantage, the adaptation of a model for a given TD must be repeated whenever the TD changes. In an acoustic monitoring scenario, such a re-adaptation of a sound event detection (SED) model would be necessary whenever the microphone of an acoustic sensor or its recording location changes. Furthermore, model-based DA requires a substantial amount of SD and TD data to avoid overfitting of the model to the SD data instead of a generalization towards TD data.

In this work, we focus on feature-based unsupervised DA methods, which aim to align the distributions of SD and TD data by applying different normalization techniques. We study two relevant acoustic monitoring scenarios, that are influenced by microphone mismatch conditions, the classification of metal ball coating materials and the classification of acoustic scenes. In particular, we first study different variants of Z-score normalization and then compare them against more sophisticated methods such as relaxed instance frequency-wise normalization (RFN) and feature projection-based unsupervised domain adaptation (FPDA). As a main contribution, we propose a modification of the original FPDA configuration, which provides a significant performance gain and even outperforms all other normalization techniques in the acoustic scene classification task.

II. Related Works

Domain adaptation is directly related to transfer learning, where knowledge learned in one domain is used to improve performance in a different but related domain [1]–[3]. In

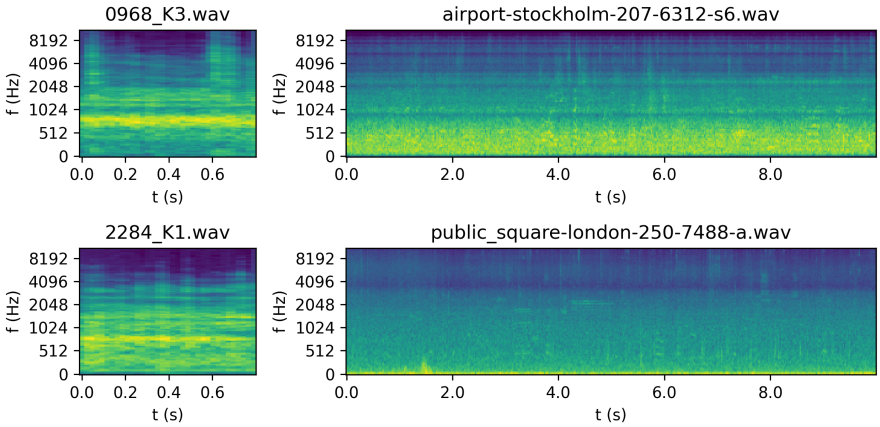


Fig. 1: Two examples taken from the IDMT-ISA-METAL-BALLS dataset (left column) and the TAU Urban Acoustic Scenes 2020 Mobile dataset (right column). Original file names are provided as titles. Log-magnitude Mel-spectrograms (compare Section IV-A) are shown to illustrate common spectral characteristics in both datasets.

general, many domain adaptation and domain generalization methods have been adapted from computer vision to audio. Several general surveys [4]–[6] group existing domain adaptation methods into categories of approaches such as adversarial learning, self-training, self-supervised learning, as well as unsupervised learning. In the audio domain, the development of domain adaptation methods is increasingly becoming an important focus of research. This is supported by the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge tasks on the topic of acoustic scene classification under mismatched recording devices, which has been carried out annually since 2018.

The first group of unsupervised domain adaptation methods investigates how specific neural network architectures such as Convolutional Neural Networks (CNNs) [7], Residual Networks (ResNets) [8], [9], and attention mechanism [10], as well as specific loss functions such as focal loss [9] help to reduce domain shift. A second group applies data augmentation, transformation, and normalization techniques, which are model-agnostic as they modify data in the feature space [11]. In machine learning, Z-score normalization (data standardization) is a standard technique to ensure that different variables have the same scale. To reduce the domain shift, Johnson and Grollmisch [12] found that it is beneficial to compute and apply first-order and second-order statistics separately in the source and target domain, a finding that we confirm in the experiments presented in this paper. A related method entitled bandwise statistics matching (BWSM) [13] aligns the mean and standard deviation of individual spectral bands between the source and target domains while preserving their original spatial structures. The Correlation Alignment (CORAL) method [14] aligns both domains by minimizing the discrepancy between the corresponding data covariances. The Deep CORAL [15] uses the same approach but instead aligns the second-order statistics of intermediate feature representations within deep neural networks.

Instead of aligning the data of the source and target domains, the Feature Projection-Based Unsupervised Domain Adaptation (FPDA) algorithm [16] projects data from both domains into a new shared subspace, which is learned to minimize the domain discrepancy. Roy et al. [17] align feature distributions across domains using feature decorrelation (whitening) and a consensus loss term, which encourages consistency across domains and domain-invariant representations. In the Fourier domain Adaptation (FDA) method [18], the data is first transformed using the Fourier transform before the Fourier coefficients are aligned between domains by matching the frequency components.

III. Datasets

In this paper, we use two datasets, for which we show in Figure 1 two example Mel-spectrograms to illustrate typical spectral characteristics. The IDMT-ISA-METAL-BALLS dataset (`MB`) is a data set of 2832 audio files from three different classes with a total duration of more than 18 minutes [19]. Each class corresponds to the sound of metal balls with a specific type of coating. The domain shift results from different microphone positions, which are recorded as four additional sets of measurements, which are referred to as “variation sets”.

The TAU Urban Acoustic Scenes 2020 Mobile dataset (`ASC`) was used in the DCASE 2020 Challenge task ”Acoustic Scene Classification with Multiple Devices” and includes a total of 26101, 10s long recordings of 10 acoustic scene classes, which were recorded in 12 European cities using four different devices. The domain shift results mainly from the different characteristics of the recording devices. Furthermore, 11 synthetic devices were simulated applying different impulse responses and dynamic range compression. We only consider the development set in this work. In this study, we split each dataset into a source domain dataset and several target domain datasets to evaluate the effectiveness of different unsupervised domain adaptation methods. In contrast to the official dataset split, we use all recordings from device A as the training

TABLE I: Feature extraction parameters for both datasets.

Parameter	MB	ASC
FFT size (samples)	2048	2048
Window size (samples)	1024	2048
Hop size (samples)	512	1024
Patch size (s)	0.4	10
Patch overlap (s)	0.4	10

set (source domain dataset) and all recordings from the real devices B and C as well as the synthetic devices S1 to S6 as individual target domains and thus use a clean data split on the device level. One should note that the dataset includes significantly more recordings from device A (14400) than from the other devices (1080 recordings each). Figure 1 illustrates two example files taken from each dataset illustrated as log-magnitude Mel-spectrograms and clearly demonstrates that audio files from both datasets have different spectral characteristics.

IV. Methodology

A. Feature Extraction

In this paper, we process audio recordings sampled at 44.1 kHz with Mel spectrograms with 128 Mel bands. We apply logarithmic magnitude scaling to reduce the overall dynamic range between foreground and background sounds. We use different feature extraction parameters for the MB and the ASC datasets introduced in Section III. Table I specifies the FFT size, window size, and hop size values used to compute Mel-spectrograms as well as the size and overlap of the processed spectral patches as input to CNN models used for both datasets.

B. Normalization Methods

In the following, we consider a feature tensor $X \in \mathbb{R}^{B \times F \times T \times C}$ of B spectrogram segments (*batch dimension*) each having F frequency bins (*frequency dimension*) and T time frames (*time dimension*). The final dimension captures C channels. While for natural images three channels are used encode the pixel intensities in the red, green, and blue color channels, we use only one channel ($C = 1$) to store magnitude spectrograms.

1) Z-score Normalization: Z-score normalization (standardization) is a common technique to scale data to have zero mean and unit variance (ZMUV). Formally, a vector $x \in \mathbb{R}^N$ is normalized using Equation 1:

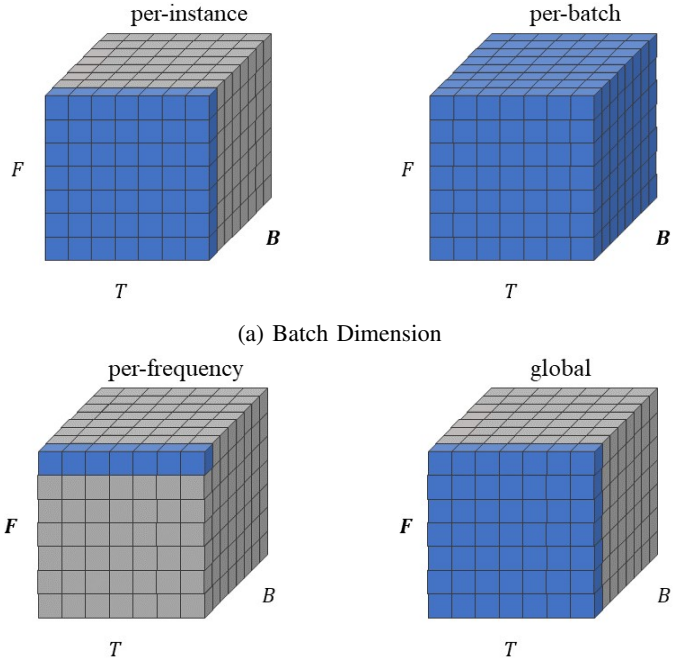
$$x_i \leftarrow \frac{x_i - \mu}{\sigma^2 + \epsilon} \tag{1}$$

where μ and σ are the mean and standard deviation of x, respectively, which can be calculated as

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{2}$$

and

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)} \tag{3}$$



(a) Batch Dimension

(b) Frequency Dimension

Fig. 2: Possible dataset partitions along the (a) batch dimension and the (b) frequency dimension.

for $i \in [1 : N]$ (denoting all integers between 1 and N) and $\epsilon < 10^{-7}$. When applying the Z-score normalization, the required statistics (μ, σ) can be computed globally on X or locally on specific partitions thereof. In this work, we investigate different partitions with respect to frequency and batch dimension as shown in Figure 2a and Figure 2b, respectively. In this work, we do not consider partitions of the data set along the time dimension.

In the batch dimension, we distinguish between a separate normalization of each instance $X_{i,:,:,:}$ with $i \in [1 : B]$ ("per-instance", Figure 2a, left) and normalization over the entire number of instances of X in the batch ("per-batch", Figure 2a, right). In the frequency dimension, we compare a normalization over each individual frequency bin $X_{:,j,:,:}$ with $j \in [1 : F]$ ("per-frequency", Figure 2b, left) and a normalization over the entire frequency range ("global", Figure 2b, right). As a third degree of freedom, we compare one scenario, where the statistics are computed and applied separately *within* each domain, and another scenario, where the statistics computed in the source domain are shared *between* domains (see Figure 3).

2) Relaxed Instance Frequency-wise Normalization (RFN): In the realm of computer vision, normalization is typically applied globally, either per batch or per instance. However, when dealing with audio processing, the frequency dimension plays a paramount role, allowing for the possibility of normalizing the data in a frequency-wise manner. In this paper, we study a frequency-wise normalization method proposed by Kim et al. [20] which can effectively address the instance-

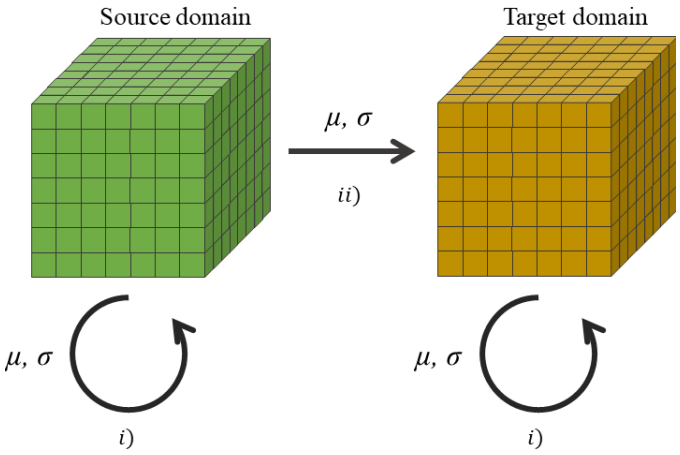


Fig. 3: Two scenarios on how the statistics (μ, σ) are computed and applied for Z-score normalization: i) within domains and ii) between domains.

TABLE II: Experiment configurations with the corresponding normalization methods and dataset partition approach.

Configuration	**Normalization Method**	**Partitioning Dimension**		
		Batch (B)	**Frequency** (F)	**Domain**
Base	-	-	-	
Z-IGW	Z-score	instance	global	within
Z-IFW	Z-score	instance	frequency	within
Z-BGW	Z-score	batch	global	within
Z-BGB	Z-score	batch	global	between
Z-BFW	Z-score	batch	frequency	within
Z-BFB	Z-score	batch	frequency	between
R-IFW	RFN	instance	frequency	within
R-BFW	RFN	batch	frequency	within
F-IFW	FPDA (**IFPDA**)	instance	frequency	within
F-BFW	FPDA	batch	frequency	within
F-BFB	FPDA	batch	frequency	between

specific domain differences present in an audio feature, while simultaneously mitigating the unintended loss of valuable discriminative information.

Algorithm 1 FPDA

$X^{F,B*T} \leftarrow reshape(moveaxis(X^{B,F,T}))$
$\mu^{F,1} \leftarrow mean(X^{F,B*T})$
$\sigma^{F,1} \leftarrow std(X^{F,B*T})$ ▷ standard deviation
$\bar{X}^{F,B*T} \leftarrow (X^{F,B*T} - \mu^{F,1}) \,/\, (\sigma^{F,1})$
$\bar{X}^{B,T*F} \leftarrow reshape(moveaxis(reshape(\bar{X}^{F,B*T})))$
$Cov^{F*T,F*T} \leftarrow covariance(\bar{X}^{B,T*F})$
$V^{F*T,F*T} \leftarrow eigenvector(Cov_s^{F*T,F*T})$

3) Feature Projection-Based Unsupervised Domain Adaptation (FPDA): In addition to the Z-score normalization and the RFN method, we study the FPDA algorithm [16]. The core of this approach involves projecting the spectrotemporal features extracted from both the source and target domains onto the principal subspace. This subspace is defined by the eigenvectors of the sample covariance matrix derived from the training data from the source domain as presented in Algorithm 1. One can do the training either using $\bar{X}_{\text{sub}}^{B,F*T}$ which is the normalized features projected on the subspace (Equation 4):

$$\bar{X}_{\text{sub}}^{B,F*T} = \bar{X}^{B,T*F} \cdot V^{F*T,F*T} \tag{4}$$

or projection of $\bar{X}_{\text{sub}}^{B,F*T}$ back into original source domain shown by $\hat{X}^{B,T,F}$ and calculated with Equation 5:

$$\hat{X}^{B,T,F} = reshape(\bar{X}_{\text{sub}}^{B,F*T} \cdot transpose(V^{F*T,F*T})) \tag{5}$$

In addition to the original FPDA method, we experimented with two other modifications in this paper.

In the initial modification, our objective was to perform within-domain normalization. To achieve this, we utilized the projection of spectro-temporal features extracted from each domain onto the principal subspace. This subspace is defined by the eigenvectors of the covariance matrix specific to the same domain.

In the second modification, we intended to apply the method to each instance individually. Consequently, we normalized each instance separately, but instead of computing the source or target subspace, we simply employed the normalized instance as subspace. The complete algorithm of Instant FPDA (IFPDA) is provided in Algorithm 2. Here $V_{\text{i}}^{F,F}$ consists of F eigenvectors of the covariance of normalized feature $Cov_{\text{i}}^{F,F}$, to map the values of each frequency band in the direction of greatest variance. $\bar{X}_{\text{sub}}^{T,F}$ is The projection of input to instant subspace, and $\hat{X}_{\text{i}}^{T,F}$ which is used for training the model is its projection back to original space. The superscripts show the dimensions of the variables.

Algorithm 2 Instant FPDA (IFPDA)

for ${x_i}^{T,F} = X_{i,1:T,1:F,:}$ `with` $i \in [1:B]$ **do**
$x_i^{F,T} \leftarrow transpose(x_i^{T,F})$
$\mu_i^{F,1} \leftarrow mean(x_i^{F,T})$
$\sigma_i^{F,1} \leftarrow std(x_i^{F,T})$ ▷ standard deviation
$\bar{X}_i^{F,T} \leftarrow (x_i^{F,T} - \mu_i^{F,1}) \,/\, (\sigma_i^{F,1})$
$Cov_i^{F,F} \leftarrow covariance(\bar{X}_i^{F,T})$ ▷ covariance of $\bar{X}_i^{F,T}$
$V_i^{F,F} \leftarrow eigenvector(Cov_i^{F,F})$ ▷ eigenvectors of $Cov_i^{F,F}$
$\bar{X}_{\text{sub}}^{T,F} \leftarrow transpose(\bar{X}_i^{F,T}) \cdot V_i^{F,F}$
$\hat{X}_i^{T,F} \leftarrow \bar{X}_{\text{sub}}^{T,F} \cdot transpose(V_i^{F,F})$
end for

C. Neural Network Architectures & Training Procedure

We apply two different CNN architectures in this paper as illustrated in Table III: For the three-class scenario given in the

TABLE III: Summary of neural network architectures for CNN model [19] (591,389 parameters) and CNN420 model [21] (799,050 parameters).

Network Block	Layers/Parameters
CNN Model	
Conv. Block 1	6 filters (3*3)
	ReLU activation
	MaxPooling2D
Conv. Block 2	18 filters (3*3)
	ReLU activation
	MaxPooling2D
Dense Block	Dense (128 neurons)
	ReLU activation
	Dropout (rate: 0.5)
Output	Dense (number of classes)
	Softmax activation
CNN420 Model	
Conv. Block 1	64 filters (5*5)
	ReLU activation
Residual Block 1	64 filters (3*1)
	Dropout (rate: 0.1)
	AvgPooling2D (2*2)
Residual Block 2	64 filters (3*3)
	Dropout (rate: 0.1)
Residual Block 3	128 filters (3*1)
	Dropout (rate: 0.1)
	AvgPooling2D (2*2)
Residual Block 4	128 filters (3*1)
	Dropout (rate: 0.1)
	Global Average Pooling
Output	Dense (number of classes)
	Softmax activation

MB dataset, we use the small-scale CNN model [12]. It includes two convolutional blocks, each having a convolutional layer, a ReLU activation function, and a max-pooling operation. The resulting feature maps are flattened and connected to two dense layers with 128 and 2 neurons, respectively. The network also incorporates a dropout regularizer with a rate of 0.5 to mitigate overfitting. For the more challenging 10-class scenario imposed by the ASC dataset, we apply the larger CNN420 model, which has been used in [21]. The model uses an initial convolutional block followed by four residual blocks, which include dropout layers with a low rate of 0.1. The final feature maps are aggregated using an average global pooling operation and processed by one dense layer. In both models, we have disabled batch normalization in the latent layers to exclusively focus on studying various normalization configurations applied to the input data.

For both the Convolutional Neural Network (CNN) and the CNN420 architecture, a learning rate of 0.001 was employed, and the optimization method adopted was Adam. Furthermore, a consistent batch size of 32 was utilized for both networks, and the training process for each was done over 150 epochs, with early stopping implemented after 70 epochs.

D. Experimental Procedure

All experiments are carried out separately on the MB and the ASC dataset. For each individual experimental configuration, we first normalize the training and validation data, then use both to train the classifier and finally normalize the target domain data before evaluating the model predictions thereof. As evaluation metrics, we measure the accuracy A_{S} in the test set of the source domain data and the accuracy A_{T} for each data set from the target domain. More precisely, we average the model performance over all instances of each variation set of the MB dataset and over all recordings of each device of the ASC dataset except for those of device A. For each data set, we perform 12 different configurations as listed in Table II. Each configuration is defined by the applied normalization method (Z-score normalization, RFN, or FPDA) listed in the second column and the data set partitioning strategy listed in the remaining columns.

V. Results

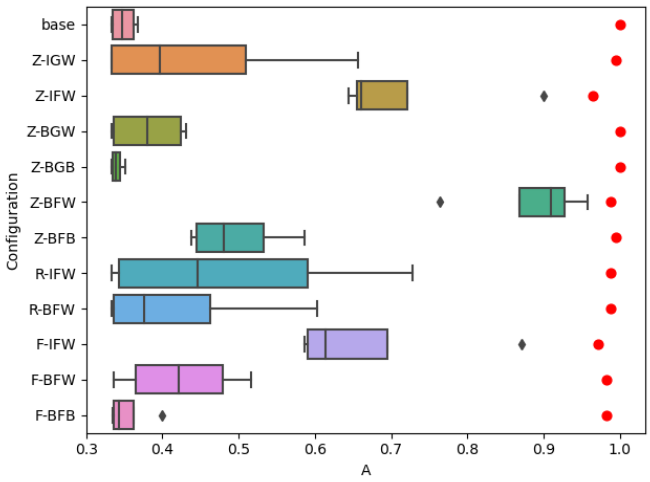
A. Normalization Methods

Figure 4 illustrates the source domain accuracy A_{S} and the average target domain accuracy A_{T} for all the configurations described in Table II and for both data sets. For A_{T}, a boxplot indicates how strong the performance varies between different target domains.

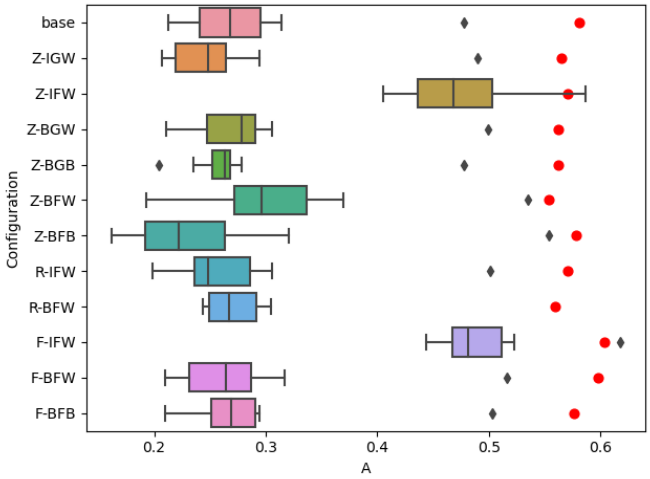
We make the following dataset-specific observations: For the MB dataset, the Z-BFW configuration (per-batch, per-frequency, within-domain Z-score normalization) clearly outperforms all other configurations with the highest accuracy score of $A_{\mathrm{T}} = 0.96$ on variation dataset 4 and average of $A_{\mathrm{T}} = 0.89$ over all variation datasets. This indicates that for a sound classification task such as imposed by the MB dataset, a rather simple Z-score normalization performed separately for each frequency band over the entire dataset leads to the best performance. A similar configuration (Z-IFW), where normalization is performed per-instance instead, achieves the second-best average performance of $A_{\mathrm{T}} = 0.72$.

For the ASC dataset, we observe a similar performance with two important differences: First, the per-instance normalization performs better (Z-IFW, $A_{\mathrm{T}} = 0.47$), which we assume is due to the much longer duration of 10s compared to 0.4s in the MB data set. Second, the newly proposed IFPDA method which is a modification to FPDA normalization with configuration F-IFW, achieves the highest average accuracy score of $A_{\mathrm{T}} = 0.49$ over the target domains.

Two more general findings need to be discussed. First, our results provide clear evidence for domain shift in both data sets, since the source domain accuracy A_{S} is consistently higher than the averaged target domain accuracy A_{T}. As expected, the domain shift varies between different variation



(a) MB dataset.



(b) ASC dataset.

Fig. 4: Source domain accuracy A_{S} (red dot) and target domain accuracy values (boxplot) with A_{T} being the average value. Results are shown for different experimental configurations based on the MB dataset (top) and the ASC dataset (bottom).

sets (MB dataset) and recording devices (ASC dataset). Second, the within-domain approach of computing the normalization parameters and applying the normalization separately in the source and target domain consistently shows higher accuracy scores compared to the between-domain. This is in line with the best performance of the "adaptive normalization" method reported by Johnson and Grollmisch [12].

B. Impact of Partition Dimensions

Given a large number of different experimental configurations, our aim is to assess the general influence of the three dimensions in the partitioning of the data set, i. e., batch, frequency, and domain, on the performance of the model

TABLE IV: Results of two-sided t-test on MB and ASC datasets. Statistical significant results ($p < 0.01$) are written in bold font.

Dimension	MB		ASC	
	t	p	t	p
Batch (B)	1.84	0.071	**4.08**	**0.0001**
Frequency (F)	**2.78**	**0.008**	1.99	0.049
Domain	2.56	0.013	2.24	0.028

independently from the normalization method. Therefore, we conducted a two-sample t-test to compare the accuracy A_{T} in the sets of configurations defined based on each of the three dimensions. The results are summarized in Table IV. We observed a significant difference in a_{T} between the groups of configurations based on different frequency dimension partitions ($t(42) = 2.78$, $p = 0.008$) for the MB dataset and between the groups of configurations based on different batch dimension partitions ($t(86) = 4.08$, $p = 0.0001$) for the ASC dataset. To summarize, these results indicate that i) a separate normalization per frequency band is important for sound classification tasks such as imposed by the MB dataset and ii) a per-instance normalization is beneficial when longer spectrogram segments are used for tasks such as in the ASC dataset.

VI. Conclusion

In this paper, we study different unsupervised methods for feature space domain adaptation. Due to their relevance for acoustic monitoring scenarios, we compare a sound classification and an acoustic scene classification use case to work out task-specific differences and particularities. For the MB dataset, our results confirm the finding of previous studies that a Z-score normalization per-frequency band and separately within-domain shows the best performance. For the ASC dataset, a newly proposed configuration of the FPDA algorithm applied per-instance and per-frequency bands separately within the source and target domains showed the best performance for both the source and target domains. A two-sided t-test confirmed the importance of the frequency and batch dimension for the data set partitioning prior to normalization. We consider our results to be easily applicable as best practices for various acoustic monitoring scenarios since the feature space domain adaptation methods are generally agnostic to the applied deep neural network architecture. From Figure 4, one can notice that the performance of the trained model on target domains for most methods was similar to the baseline model or other words not using any normalization at all. This shows that normalization or any data processing method, in general, should be relevant to the data and the task and should be done in a way that highlights the important features, which for audio data as already discussed are per frequency and per instance.A possible research question for future work is how

the integration of feature-space and latent-space could enhance the efficacy of domain adaptation techniques, or alternatively, how it might prove useful in addressing issues related to domain shift.

VII. Acknowledgements

The authors would like to thank Franca Bittner and Sascha Grollmisch for detailed discussion. This study was supported by the German Research Foundation (AB 675/2-2) as well as internal funding by Fraunhofer society.

References

[1] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

[3] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[4] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual Domain Adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[5] G. Csurka, "A Comprehensive Survey on Domain Adaptation for Visual Applications," in *Domain Adaptation in Computer Vision Applications*, ser. Advances in Computer Vision and Pattern Recognition, G. Csurka, Ed. Cham: Springer International Publishing, 2017, pp. 1–35.

[6] M. Wang and W. Deng, "Deep Visual Domain Adaptation: A Survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[7] Y. Shao, X. Ma, Y. Ma, and W.-Q. Zhang, "Thuee submission for DCASE 2020 challenge task1a," DCASE2020 Challenge, Tech. Rep., June 2020.

[8] S. Suh, S. Park, Y. Jeong, and T. Lee, "Designing acoustic scene classification models with CNN variants," DCASE2020 Challenge, Tech. Rep., June 2020.

[9] W. Gao and M. McDonnell, "Acoustic scene classification using deep residual networks with focal loss and mild domain adaptation," DCASE2020 Challenge, Tech. Rep., June 2020.

[10] L. Jie, "Acoustic scene classification with residual networks and attention mechanism," DCASE2020 Challenge, Tech. Rep., June 2020.

[11] W. M. Kouw and M. Loog, "A Review of Domain Adaptation without Target Labels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 766–785, 2021.

[12] D. Johnson and S. Grollmisch, "Techniques improving the robustness of deep learning models for Industrial Sound Analysis," in *Proceedings of the 2020 European Signal Processing Conference (EUSIPCO)*, Online, 2021, pp. 81–85.

[13] A. I. Mezza, E. A. P. Habets, M. Müller, and A. Sarti, "Unsupervised Domain Adaptation for Acoustic Scene Classification Using Band-Wise Statistics Matching," in *Procedings of the 2020 28th European Signal Processing Conference (EUSIPCO)*, Amsterdam, The Netherlands, 2021, pp. 11–15.

[14] B. Sun, J. Feng, and K. Saenko, "Return of Frustratingly Easy Domain Adaptation," *Proceedings of the 13th AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, pp. 2058–2065, 2016.

[15] B. Sun and K. Saenko, "Deep CORAL: Correlation Alignment for Deep Domain Adaptation," in *Computer Vision – ECCV 2016 Workshops*, G. Hua and H. Jégou, Eds. Cham: Springer International Publishing, 2016, vol. 9915, pp. 443–450.

[16] A. I. Mezza, E. A. P. Habets, M. Müller, and A. Sarti, "Feature Projection-Based Unsupervised Domain Adaptation for Acoustic Scene Classification," in *Proceedings of the 2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, Espoo, Finland, 2020, pp. 1–6.

[17] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulo, N. Sebe, and E. Ricci, "Unsupervised Domain Adaptation Using Feature-Whitening and Consensus Loss," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, 2019, pp. 9463–9472.

[18] Y. Yang and S. Soatto, "FDA: Fourier Domain Adaptation for Semantic Segmentation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, 2020, pp. 4084–4094.

[19] S. Grollmisch, J. Abeßer, J. Liebetrau, and H. Lukashevich, "Sounding industry: Challenges and datasets for Industrial Sound Analysis," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, A Coruña, Spain, 2019, pp. 1–5.

[20] 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 *Proceedings of the INTERSPEECH conference*, Incheon, Korea, 2022, pp. 2393–2397.

[21] S. Grollmisch and E. Cano, "Improving Semi-Supervised Learning for audio classification with FixMatch," *Electronics*, vol. 10, no. 15, 2021.