INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Background Domain Switch: A Novel Data Augmentation Technique for Robust Sound Event Detection

Wei-Cheng Lin[1], *Luca Bondi*[2] *and Shabnam Ghaffarzadegan*[2]

[1]The University of Texas at Dallas, Richardson, USA
[2]Robert Bosch Research and Technology Center, USA - Bosch Center for Artificial Intelligence

wei-cheng.lin@utdallas.edu, luca.bondi@us.bosch.com, shabnam.ghaffarzadegan@us.bosch.com

Abstract

Data augmentation is a key component to achieve robust and generalizable performance in *sound event detection* (SED). A well trained SED model should be able to resist the interference of non-target audio events and maintain a robust recognition rate under unknown and possibly mismatched testing conditions. In this study, we propose a novel *background domain switch* (BDS) data augmentation technique for SED. BDS utilizes a trained SED model on-the-fly to detect backgrounds in audio clips, and switches them among the data points to increase sample variability. This approach can be easily combined with other types of data augmentation techniques. We evaluate the effectiveness of BDS by applying it to several state-of-the-art SED frameworks, and used both publicly available datasets as well as a synthesized mismatched test set. Experiment results systematically show that BDS obtains significant performance improvements from all evaluation aspects. The code is available at: https://github.com/boschresearch/soundsee-background-domain-switch

Index Terms: sound event detection, data augmentation, non-target audio events

1. Introduction

Sound event detection (SED) systems aim at describing sounding objects by detecting, categorizing, and locating the time boundaries of acoustic events (”what” and ”when”) in a stream or an audio file [1]. With the huge success of *deep learning* (DL) approaches in several speech-related tasks, such as automatic speech recognition [2] and speech enhancement [3], recent SED systems based on DL [4] are showing significant improvements over handcrafted features fed to machine-learning algorithms. However, DL models are prone to overfitting and may exhibit poor generalization, especially when there is a lack of diverse and large training samples [5]. Therefore, model regularization techniques such as dropout [6], or data augmentation strategies [7] are critical to achieve robust recognition performance. Using appropriate regularization in an SED task proves crucial, since it is hard to collect diverse and large high-quality timestamped labeled audio data (i.e., strong labels) [8, 9, 10].

Besides the DL modeling perspective, the performance of SED is notably affected by interference of non-target audio events (i.e., other background sounds that are not included in the SED recognition classes). The existence of non-target events in the testing stage could severely degrade the model performance [11, 12]. For instance, Ronchini *et al.* [11] found that simply incorporating non-target audio events in the training data can improve the system accuracy. However, this approach might not be feasible in realistic settings given the unknown nature of non-target audio events. An SED model robust to non-target events could avoid undesired false alarms, a property important in many applications to improve user experience [13]. Therefore, it is necessary to assess the SED models for both target and non-target testing environments, to obtain a comprehensive evaluation.

Motivated by this, we propose a novel *background domain switch* (BDS) data augmentation to improve the robustness of SED models in presence of non-target audio events. BDS firstly relies on a self-labeling process to automatically identify background sounds (i.e., non-target background audio events) in given audio clips. Next, it randomly switches these backgrounds among batch of data samples. This method produces an augmented training set without altering the original labels (i.e., target event classes). BDS can be applied along with other data augmentations techniques such as MixUp [14], SpecAug [7], etc., to further increase the model regularization. In this study, we adopt the conventional DCASE 2022-Task4 [15, 16] setup to evaluate our approach. In addition, we carefully curate a synthesized *mismatched* test set to evaluate model performance under the presence of non-target audio events interference. Our experimental results show that BDS leads to statistically significant improvements over other baselines under *polyphonic sound detection score* (PSDS) [13] metrics. We also demonstrate that BDS is flexible enough to combine with other advanced modeling approaches such as *frequency dynamic convolution* framework (FDY-SED) [9], to further improve the model performances. In summary, the main contributions of this work are: 1) A novel BDS data augmentation approach to improve SED models robustness in the presence of non-target audio events. 2) A method that is flexible to be integrated into other existing SED frameworks and regularization techniques for better recognition performance.

2. Background

2.1. Sound Event Detection

Various DL techniques were proposed to tackle the limited availability of strongly labeled audio data necessary for building SED models. For instance, the *Mean-Teacher* (M-T) semi-supervised training scheme was proposed to leverage large amount of unlabeled data to extract additional complementary information [17, 18]. McFee *et al.* [19] treated SED as a *multiple instance learning* problem, where the model implicitly infers the temporal boundary of events from a given static clip-level label (i.e., weak labels without time boundaries) by utilizing adaptive pooling operators. Ebbers and Haeb-Umbach [20] explicitly exploited the model predictions on the unlabeled and weakly-labeled data as their pseudo-strong labels, which are then utilized to retrain the model. Another main research direc-

This work was done as research intern at Bosch Research and Technology Center, Sunnyvale, CA, USA.

10.21437/Interspeech.2023-176

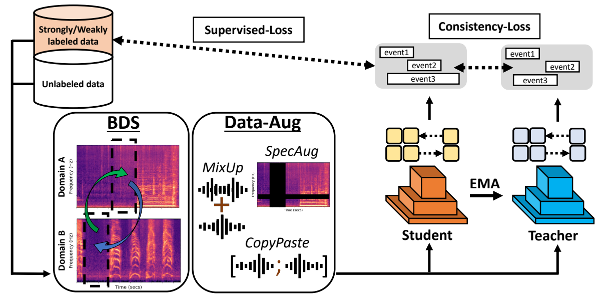


Figure 1: *Overview of the SED framework with proposed BDS module, EMA: Exponential Moving Average.*

tion in SED is designing audio event specific features or models. For instance, events with different temporal properties (e.g., cat meowing as short event) or distinct frequency patterns (e.g., stationary high-frequency vacuum cleaner sound) can be modeled better by dynamic-kernel convolution layers such as *selective kernel* (SK) or FDY-SED framework [18, 21, 9, 22]. In our work, we follow the standard M-T framework from [17] to build an SED model (Sec. 3.1). In addition, we introduce BDS into the existing FDY-SED framework [9] to show the adaptability of our approach (Sec. 5.1) into existing techniques.

2.2. Data Augmentation Techniques in Audio Domain

Data augmentation techniques are of paramount importance to improve the performance of DL models under data-limited scenarios. In audio/speech domains, many tasks including SED rely heavily on different types of augmentation approaches [9, 18, 17]. The most straightforward method is to manipulate the input acoustic features, with noise injection [23], frame-shift, time/frequency warping and masking (i.e., SpecAugment [7]) or more advanced regional frequency amplification/reduction (i.e., FilterAugment [10]). Another conventional approach is focusing on the generation of new training samples. For instance, SCAPER [24] synthesizes different background tracks with the target foreground sounds (e.g., acoustic events) to create unlimited synthetic datasets. MixUp [14] and CopyPaste [25] perform on-the-fly addition or concatenation of two data samples that are randomly selected in a training batch, resulting in new data points. Different from above-mentioned, our BDS approach *switches* background features that are detected by the SED model among the training batch to produce augmented samples.

3. Proposed method

3.1. System Framework

We directly adopt the official baseline framework[1] from DCASE2022 Challenge as our backbone SED model. Figure 1 depicts an overview of the framework. The model uses the *Mean-Teacher* (M-T) training strategy from [26] to leverage a mix of strongly labeled, weakly labeled, and unlabeled data. Both student and teacher models are fed with augmented samples, according to the augmentation strategy at hand. A consistency loss (i.e., mean squared error, MSE) is imposed to push the predictions from student and teacher models to be as similar as possible. A supervised loss (i.e., binary cross-entropy, BCE) is applied to the predictions of the student model, frame-wise when for strongly-labeled samples, clip-wise for weakly-labeled samples. The teacher model updates its weights from the student model through an *exponential moving average* (EMA) strategy. An attention-based pooling layer is applied to summarize frame-level predictions into the clip-level results. The model architecture is a standard *convolutional recurrent neural network* (CRNN). More details about the model can be found in [26].

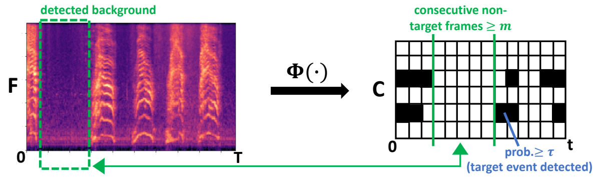


Figure 2: *Visualization of the background detection process in BDS using on-the-fly trained SED model* $\Phi(\cdot)$*. Variables T, F and t, C refer to the input feature length, dimension and predicted output sequence length, classes, respectively.*

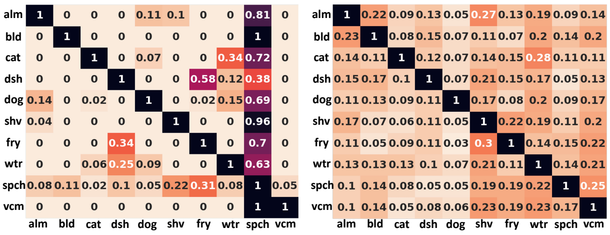
3.2. Background Domain Switch

The core component of BDS is leveraging the trained SED model on-the-fly during the training process as a background-foreground recognizer. Here, we refer to background as any sounds (including silence and noises) that does not belong to any *target event* in the SED model. More specifically, let $\mathbf{Y} = \{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_t}\} \in \mathbb{R}^{\mathbf{t}\times\mathbf{C}}$ represents the sequential prediction output of $\Phi(\mathbf{X})$, where C is the number of target classes, t is the number of time frames in $\mathbf{Y}$, $\Phi(\cdot)$ denotes the student model from the previous training iteration, and $\mathbf{X} \in \mathbb{R}^{T\times F}$ is the input 2D acoustic feature map with T frames and F features. Note that $t < T$ if any presence of temporal-wise pooling layers in the convolutional encoder of the SED model. A background segment is detected between frames t_1 and t_2 if $\mathbf{y_i(j)} < \tau,\ \forall \mathbf{i} \in [\mathbf{t_1}, \mathbf{t_2}],\ \mathbf{j} \in [\mathbf{1}, \mathbf{C}]$ and $t_2 - t_1 \geq m$, i.e. a background segment is a period of at least m consecutive frames where no class is predicted with confidence above τ.

Figure 2 shows an example of the background detection process for BDS. The two factors τ and m control the amount and confidence level of the detected background segments. For instance, we can reduce τ to increase the sensitivity to target events, and thus reduce the contamination of background segments. Or we can increase m, to prevent short segments misclassified as background to enter the pool of background segments. Besides τ and m, it is also crucial to incorporate BDS only at a later stages of training, i.e. once the $\Phi(\cdot)$ has gained enough discrimination ability to serve as a reliable background detector. In general, we apply BDS only to the last x% training epochs. Note that in case of availability of strong labels in the training data, we can directly use the ground-truth to identify background segments. The detected background segments are randomly selected and switched within a mini-batch, thus generating on-the-fly augmented data, without changing the original event labels. In case of different duration between background segments, we simply crop segments that are too long, or repeat over time those that are too short.

BDS is a flexible augmentation technique that can be applied to different types of training conditions. For example, when having mixed training data from different domains (e.g., synthetic versus real-world data), having training samples in the same domain but from different datasets (e.g., cross-corpus training), or simply having data within same training set. Another interesting property of BDS is that it is direction-specific depending on the application scenario of the SED system. For instance, when having a mixed of synthetic and real-world data we might not expect the backgrounds from the synthetic data to

[1] `https://github.com/DCASE-REPO/DESED_task/tree/master/recipes/dcase2022_task4_baseline`



(a) Public-Eval. Test (b) Mismatch Test

Figure 3: *Label co-occurrence ratio across events for the two blind test sets.*

appear in the real life scenarios. Hence, we perform unidirectional BDS to only switch synthetic backgrounds with the real-world ones. For the unknown or mismatch test cases, we can perform *bi-directional BDS* (Bi-BDS) to switch backgrounds in all the training samples to further improve the overall generalization performance (see Sec. 5.1).

4. Resources and Experimental Settings

4.1. Datasets

Domestic environment sound event detection (DESED) dataset [8] is used for the DCASE2022-Task4, which consists of weakly-labeled (1,578 clips), synthesized strongly-labeled (10,000 clips) and unlabeled (14,412 clips) training sets. Additional *validation* (Dev. Test) set (1,168 clips) is provided for hyperparameters tuning and model development. The final model performances are compared on a blind *public-evaluation* (Public-Eval. Test) set (692 clips). All the audio clips are processed to have a fixed 10 seconds duration. Synthetic data is generated using SCAPER [24], and all the other sets are collected from real-world audio clips (i.e., YouTube videos and AudioSet [27]). The 10 target sound events included in DESED are: *alarm/bell ringing* (alm), *blender* (bld), *cat* (cat), *dishes* (dsh), *dog* (dog), *electric shaver/toothbrush* (shv), *frying* (fry), *running water* (wtr), *speech* (spch) and *vacuum cleaner* (vcm). In addition, we purposely curated a synthetic *Mismatch* Test set (500 clips) with SCAPER, to evaluate the model robustness toward non-target events. We incorporated 8 outdoor ambient recordings (*construction*, *stream*, *thunderstorm*, *street*, *babble*, *bus station*, *basketball* and *baseball court*) from Airborne Sound [28] as background tracks to combine with the target foreground events from DESED.

Figure 3 shows the label distribution/co-occurrence of Public-Eval Test set of DESED (a) and the curated Mismatch Test set (b). As seen in the confusion matrix (a), the label distribution in Public-Eval Test set is far from uniform, as some events have higher co-occurrence than others. For example, speech has a strong presence throughout the dataset, and it always co-occurs with blender and vacuum sounds. Likewise, *frying* sound co-occurs 58% of the time with *dishes* sound. These co-occurrences are directly related to the acoustic environment that the data is collected in (e.g. domestic environment). Similar trends are observed in DESED training set. Unlike the Public-Eval set of DESED data, we curated the Mismatch Test set with a uniform distribution of co-occurrences across different classes. We made this design choice to evaluate the generalizability of the trained models toward different label distributions, regardless of the environmentally forced audio event co-occurrences.

Table 1: *Summary of system performance based on different data augmentation approaches. The symbols* $*$ *and* † *indicate that the performance improvements over the* Baseline *and* StrongAug *are statistically significant, respectively. Standard deviation of system performances across the 10 running trials are all below 0.02.*

Approach	Public-Eval. Test		Mismatch Test	
	PSDS-1	*PSDS-2*	*PSDS-1*	*PSDS-2*
Baseline	0.3548	0.5464	0.2180	0.3925
StrongAug	0.3536	0.5614*	0.2457*	0.4371*
Baseline +BDS	**0.3699***†	0.5546	0.2614*†	0.4263*
Baseline +Bi-BDS	0.3647*†	0.5552	0.2509*	0.4228*
StrongAug +BDS	0.3595	**0.5718***†	0.2625*†	0.4484*†
StrongAug +Bi-BDS	0.3497	0.5658*	**0.2789***†	**0.4829***†

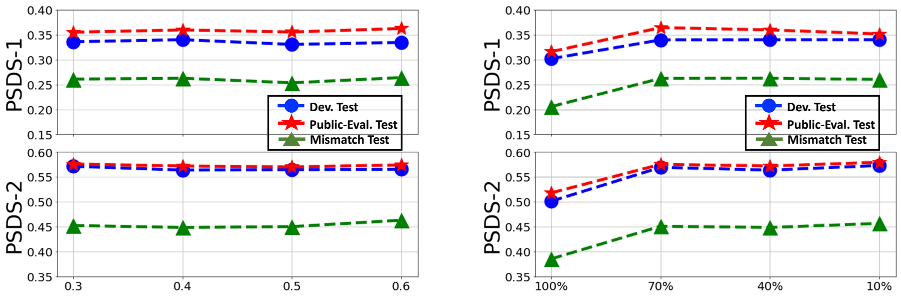
4.2. Experimental Setup

All training hyper-parameters (optimizer, learning rate, batch size, weighting of loss functions) are the same as in the official baseline. Input acoustic feature is a 128-dimensional log Mel-spectrogram, extracted from mono, 16kHz audio clips using 16ms hop size. The BDS decision threshold τ (see Section 3.2), is set to 0.4, m is set to 40 frames (i.e., 0.64 secs) and BDS is applied once training has reached 60% of the total number of epochs, (i.e. we apply BDS in the last 40% of the training epochs). We evaluate the impact of BDS hyperparameters in Section 5.2. We measure the performance under the PSDS-1 and PSDS-2 metrics [13], to evaluate the SED model for both time-localization and detection of acoustic events, respectively. We report the average results of PSDS scores after running 10 trials with different network initialization (i.e. different random seeds). We implement this strategy to conduct statistical analysis using a two-tailed t-test, where the statistical significance is defined when p-value $\leq$ 0.05. By default, the data augmentation of baseline model is applied with 50% chance to perform MixUp. Studies have shown the benefits of increasing augmentation complexity in SED model [9, 18, 21], therefore we add another *strong augmentation* (StrongAug) setting to our comparison, which has a 75% chance of performing MixUp, CopyPaste and SpecAugment.

5. Experimental Results and Analysis

5.1. System Performance Comparison

Table 1 summarizes model performances with different data augmentations under the baseline framework. There are three major points to highlight. First, *StrongAug* significantly improves the generalization performance. This result reemphasizes that increasing the data augmentation complexity is effective to build a better SED model. Second, by introducing the proposed BDS approach prior to other data augmentations, we consistently gain significant performance improvement, especially in the Mismatch Test scenario. Specifically, we can see a 13.5% and 10.5% relative performance gains after applying BDS for PSDS-1 and PSDS-2 on the Mismatch Test, respectively (comparison of *StrongAug* vs *StrongAug+Bi-BDS*). BDS switches the background features among different data sources, adding another layer of data complexity on top of other data augmentation approaches. For instance, MixUp mixes two data points based on the BDS outputs, which are blended with dif-



(a) Event Decision Threshold (τ)

(b) Last x% of Training Epochs

Figure 4: *Performance trends on the three test sets for BDS using different hyperparameters on top of* StrongAug *setting.*

Table 2: *Summary of system performance based on the FDY-SED framework [9]. The symbol $*$ indicates that the performance improvements over the original* FDY-SED *model are statistically significant. Standard deviation of system performances across the* 10 *trials are all below* 0.02.

Approach	Public-Eval. Test		Mismatch Test	
	PSDS-1	*PSDS-2*	*PSDS-1*	*PSDS-2*
FDY-SED	0.4290	0.6584	0.1464	0.2247
FDY-SED +BDS	**0.4491***	0.6686*	0.1386	0.2154
FDY-SED +Bi-BDS	0.4431*	**0.6749***	**0.1640***	**0.2789***

ferent backgrounds. This further increases model regularization during training stage and thus benefits the model performance and robustness. Third, we observe that the model performs better for Public-Eval Test case when using unidirectional BDS, while Bi-BDS is better for Mismatch Test. This interesting finding indicates that we can adjust the BDS setup depending on the application scenario of SED model. For the known and matched testing conditions, we can simply adopt unidirectional BDS for domain-specific background switch to serve as scenario enhanced data augmentation. To increase the model robustness against unknown non-target interference, Bi-BDS is a favorable option.

We also apply BDS on other advanced SED framework to further validate the generality of our proposed approach. *Frequency dynamic convolution* (FDY-SED) model [9] uses adaptable convolutional kernels to better capture intrinsic characteristics of different sound events (e.g., stationary sounds with certain frequency regions like vacuum cleaner), achieving state-of-the-art model performance in the DCASE Challenge. Frameshift, MixUp, time masking and FilterAugment are applied as the data augmentations in FDY-SED. Similar to the previous experiments, we apply BDS prior to these augmentations. All other settings such as model architecture and training parameters are the same as the released Github repository[2]. Note that we report the *average* model performance based on different trials to run statistical test, which is different from the original paper [9] that reports the *best* performance among these trials. Table 2 summarizes the results based on the FDY-SED framework. We observe significant improvements for all the evaluation scenarios after adding BDS. In particular, the relative gains reach to 12% and 24% for PSDS-1 and PSDS-2 on the Mismatch Test, respectively (comparing to *FDY-SED+Bi-BDS*). The result demonstrates that the proposed BDS approach is flexible and effective to be integrated into other advanced SED frameworks. By simply adding an additional BDS operation prior to the original data augmentation module, we can achieve better model generalization and robustness.

5.2. Impacts of BDS Hyperparameters

In this section, we examine the impact of different hyperparameters in BDS. Figure 4 illustrates the performance trends for the three test sets (Dev., Public-Eval. and Mismatch). As mentioned in Section 4.1, we use the Dev Test for BDS hyperparameters tuning. Here, we show the results of varying event decision threshold τ and number of training epochs in which BDS is applied x%. The hyperparameter m, minimum consecutive frames, has a similar flat-like trend as Figure 4a and is not shown due to space constraints. We observe that all the results have high consensus across three test sets (i.e., similar trends), and the Mismatch Test results consistently show a lower performance due to the non-target event interference. For the threshold τ, we see a flat trend across different settings, indicating a minor role of the parameter. This implies that the trained SED model has high confidence toward its predictions (i.e., predicted class probabilities are typically located at the two edges such as below 0.1 or above 0.9). Empirically, smoothing model's predictions can bring better generalization performance, thus ensemble models could provide additional benefits [21]. On the other hand, the entry epoch of BDS in the training process plays a critical role in the final model performance. A clear drop of the model performance is observed when BDS is applied early on during the training (i.e., see leftmost data point in Figure 4b, which means we apply BDS from the start of the training process). In the early stages of training, the SED model is not powerful enough to discriminate between the foreground and background events and will result in a data augmentation that further confuses the model training.

6. Conclusions

In this paper we proposed a novel *background domain switch* (BDS) data augmentation approach to improve SED models. BDS leverages the trained SED model on-the-fly as a background detector, switching different backgrounds among the training data for better model generalization and robustness. We demonstrated that BDS can be easily integrated into other existing state-of-the-art SED frameworks by simply introducing it prior to other data augmentation approaches. Furthermore, we purposely curated a mismatch test set with balanced label distribution using background sounds that are not present in the training data. This set is used to evaluate the robustness of the trained models toward mismatched condition and different label distribution. We hope the design concept of mismatch test set will inspire future research in building a comprehensive mismatch evaluation criteria. In future works, we plan to extend BDS to multi-modal learning, where we can utilize complementary information from other modalities to detect background events.

[2] https://github.com/frednam93/FDY-SED

7. References

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine (IEEE SPM 2021)*, vol. 38, no. 5, pp. 67–83, 2021.

[2] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, 2020, pp. 7829–7833.

[3] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *Interspeech 2013*, vol. 2013, 2013, pp. 436–440.

[4] N. Turpault and R. Serizel, "Training sound event detection on a heterogeneous dataset," *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2020.

[5] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning," in *International Conference on Machine Learning (ICML 2020)*. PMLR, 2020, pp. 8093–8104.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research (JMLR)*, vol. 15, no. 1, pp. 1929–1958, 2014.

[7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*, vol. 2019, 2019, pp. 2613–2617.

[8] N. Turpault, R. Serizel, J. Salamon, and A. P. Shah, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2019.

[9] H. Nam, S.-H. Kim, B.-Y. Ko, and Y.-H. Park, "Frequency dynamic convolution: Frequency-adaptive pattern recognition for sound event detection," in *Interspeech 2022*, vol. 2022, 2022, pp. 2763–2767.

[10] H. Nam, S.-H. Kim, and Y.-H. Park, "Filteraugment: An acoustic environmental data augmentation method," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022)*, 2022, pp. 4308–4312.

[11] F. Ronchini, R. Serizel, N. Turpault, and S. Cornell, "The impact of non-target events in synthetic soundscapes for sound event detection," *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021.

[12] N. Turpault, R. Serizel, S. Wisdom, H. Erdogan, J. R. Hershey, E. Fonseca, P. Seetharaman, and J. Salamon, "Sound event detection and separation: a benchmark on desed synthetic soundscapes," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, 2021, pp. 840–844.

[13] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, 2020, pp. 61–65.

[14] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations (ICLR 2018)*, 2018.

[15] A. Mesaros, A. Diment, B. Elizalde, T. Heittola, E. Vincent, B. Raj, and T. Virtanen, "Sound event detection in the dcase 2017 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 6, pp. 992–1006, 2019.

[16] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2018.

[17] L. Delphin-Poulat, R. Nicol, C. Plapous, and K. Peron, "Comparative assessment of data augmentation for semi-supervised polyphonic sound event detection," in *Conference of Open Innovations Association (FRUCT 2020)*. IEEE, 2020, pp. 46–53.

[18] X. Zheng, Y. Song, I. McLoughlin, L. Liu, and L.-R. Dai, "An improved mean teacher based method for large scale weakly labeled semi-supervised sound event detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, 2021, pp. 356–360.

[19] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2180–2193, 2018.

[20] J. Ebbers and R. Haeb-Umbach, "Self-trained audio tagging and sound event detection in domestic environments," *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2021.

[21] X. Zheng, H. Chen, and Y. Song, "Zheng ustc team's submission for dcase2021 task4–semi-supervised sound event detection," *Tech. Report, DCASE 2021 Challenge*, 2021.

[22] X. Xia, R. Togneri, F. Sohel, Y. Zhao, and D. D. Huang, "Sound event detection using multiple optimized kernels," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1745–1754, 2020.

[23] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," *Advances in Neural Information Processing Systems (NIPS 2020)*, vol. 33, pp. 6256–6268, 2020.

[24] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2017)*, 2017, pp. 344–348.

[25] R. Pappagari, J. Villalba, P. Żelasko, L. Moro-Velazquez, and N. Dehak, "Copypaste: An augmentation method for speech emotion recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, 2021, pp. 6324–6328.

[26] L. Delphin-Poulat and C. Plapous, "Mean teacher with data augmentation for dcase 2019 task 4," *Tech. Report, Orange Labs Lannion, France*, 2019.

[27] J. F. Gemmeke, D. P. 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 *IEEE international conference on acoustics, speech and signal processing (ICASSP 2017)*, 2017, pp. 776–780.

[28] P. Virostek, "Airborne Sound: Sound effects library," https://www.airbornesound.com/.