MULTI-ITERATION MULTI-STAGE FINE-TUNING OF TRANSFORMERS FOR SOUND EVENT DETECTION WITH HETEROGENEOUS DATASETS

Florian Schmid[1], Paul Primus[1], Tobias Morocutti[1], Jonathan Greif[1], Gerhard Widmer[1,2]

[1]Institute of Computational Perception (CP-JKU), [2]LIT Artificial Intelligence Lab,
Johannes Kepler University Linz, Austria
{florian.schmid, paul.primus}@jku.at

ABSTRACT

A central problem in building effective sound event detection systems is the lack of high-quality, strongly annotated sound event datasets. For this reason, Task 4 of the DCASE 2024 challenge proposes learning from two heterogeneous datasets, including audio clips labeled with varying annotation granularity and with different sets of possible events. We propose a multi-iteration, multi-stage procedure for fine-tuning Audio Spectrogram Transformers on the joint *DESED* and *MAESTRO Real* datasets. The first stage closely matches the baseline system setup and trains a CRNN model while keeping the pre-trained transformer model frozen. In the second stage, both CRNN and transformer are fine-tuned using heavily weighted self-supervised losses. After the second stage, we compute strong pseudo-labels for all audio clips in the training set using an ensemble of fine-tuned transformers. Then, in a second iteration, we repeat the two-stage training process and include a distillation loss based on the pseudo-labels, achieving a new single-model, state-of-the-art performance on the public evaluation set of DESED with a PSDS1 of 0.692. A single model and an ensemble, both based on our proposed training procedure, ranked first in Task 4 of the DCASE Challenge 2024.[1].

Index Terms— DCASE Challenge, Sound Event Detection, ATST, BEATs, PaSST, DESED, MAESTRO Real, pseudo-labels

1. INTRODUCTION

The goal of Sound Event Detection (SED) is to identify specific acoustic events and their timing within audio recordings. Reliable SED systems enable applications in numerous domains, for example, in security and surveillance [1], smart homes [2], or health monitoring [3]. A main driver of research in this field is the annual DCASE Challenge, particularly Task 4, which focuses on SED.

State-of-the-art SED systems are based on deep learning approaches, requiring a substantial amount of annotated data. Their performance is mainly limited by the lack of strongly annotated real-world sound event datasets [4]. Hence, previous editions of Task 4 focused on learning from weakly labeled data [5], semi-supervised learning strategies [6], and utilizing synthetic strongly labeled data [7]. While Task 4 has been based on the DESED dataset [7] in previous years, the key novelty of the 2024 edition is a unified setup including a second dataset, MAESTRO Real [4]. As domain identification is prohibited, the goal is to develop a single system that can handle both datasets despite crucial differences, such as labels with different temporal granularity and potentially missing labels. In fact, because of the lack of strongly annotated,

high-quality real-world data, the hope is that learning from two datasets in parallel has a synergetic effect and eventually increases the generalization performance on both datasets.

The main contributions of this work are as follows: **(1)** We introduce a multi-iteration, multi-stage training routine for fine-tuning pre-trained transformer models on SED using heterogeneous datasets. **(2)** We demonstrate that combining fine-tuned transformers – ATST [8], PaSST [9], and BEATs [10] – into a diverse ensemble to generate pseudo-labels, and using these pseudo-labels in a subsequent training iteration, significantly enhances single-model performance, yielding a relative increase of 25.9% in terms of polyphonic sound detection score [11,12] (PSDS1) on DESED and 2.7% in terms of segment-based mean partial area under the ROC curve (mpAUC) on MAESTRO, compared to the baseline system. **(3)** We conduct an ablation study to analyze the impact of the heterogeneous datasets and design choices related to them.

On DESED, we set a new state of the art on the public evaluation set, raising single-model performance from 0.686 [11] to 0.692 in terms of PSDS1. A single model and an ensemble, both based on our proposed training procedure, ranked first in the respective categories in Task 4 of the DCASE Challenge 2024 [13].

2. RELATED WORK

SED Architectures: Since the 2018 edition [14], the baseline system is based on a Convolutional Recurrent Neural Network (CRNN). A large increase in performance happened in the 2023 edition, as the baseline used BEATs [10] embeddings concatenated with CNN embeddings, which were then fed to the RNN, with a relative increase of almost 50% in PSDS1 score. Top-ranked systems in the 2023 edition improved over the baseline architecture with variations of frequency-dynamic convolution [15]. Recently, Shao et al. [16] proposed a procedure to fine-tune large pre-trained transformers on the DESED dataset with a two-stage training procedure, establishing a new state of the art. They showed that the key to avoiding overfitting is placing a large weight on the self-supervised losses to take advantage of the larger amount of unlabeled data.

Data Augmentation: As strongly annotated data is very limited, data augmentation is an important strategy to improve the generalization of SED systems. In this regard, Filter-Augment [17] simulates variations in the acoustic environment by applying different weights to frequency bands, forcing the model to extract information from wider frequency regions. Strategies for recording device generalization in Acoustic Scene Classification apply similar frequency weighting mechanisms: Frequency-MixStyle [18,19] mixes the frequency information of two audio clips in the dataset, and Device-Impulse augmentation [20] convolves an audio clip with an impulse response of a real recording device. Recently, Fre-
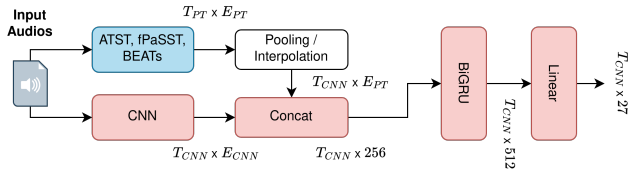

Figure 1: Overview of System Architecture. Blue are the pre-trained transformers; the red blocks together comprise the CRNN.

quency Warping [8], which stretches or squeezes spectrograms in the frequency dimension, was shown to be an integral part when fine-tuning transformers on the DESED dataset [16].

Pseudo-labels: Both of the top-ranked approaches in the 2022 and 2023 editions of Task 4 computed pseudo-labels. Ebbers et al. [21] use a multi-iteration self-training procedure in which pseudo-labels, predicted by an ensemble, are iteratively refined. Kim et al. [22] employ a two-iterations setup in which strong pseudo-labels for weakly labeled, unlabeled, and AudioSet [23] clips are computed from an ensemble of models from the first training iteration. The computed pseudo-labels are converted into hard labels and used as additional targets in a second training iteration.

3. SYSTEM ARCHITECTURE

Figure 1 gives an overview of our SED system. It consists of two independent audio embedding branches (CNN and transformer), the outputs of which are pooled to the same sequence length. A Recurrent Neural Network (RNN) derives strong predictions from these combined sequences. Compared to the baseline [13] we experiment with two additional Audio Spectrogram Transformers besides BEATs [10], namely, ATST [8] and PaSST [9]. In addition to adaptive average pooling, we experiment with linear and nearest-exact interpolation to align transformer and CNN sequence lengths. In the following, we briefly describe the transformer models used in our setup. We refer the reader to [24] for more details.

ATST-Frame [25](denoted ATST in the following) was specifically designed to produce a sequence of frame-level audio embeddings. The architecture of ATST is based on the Audio Spectrogram Transformer (AST) [26]; it is pre-trained in a self-supervised manner on AudioSet. In our experiments, we use a checkpoint of ATST that is fine-tuned on the weak labels of AudioSet.

fPaSST: The Patchout faSt Spectrogram Transformer (PaSST) [9] is an improved version of the original AST [26] that shortens the training time and improves the performance via patchout regularization. We adapt PaSST to return frame-level predictions and call the resulting model Frame-PaSST (fPaSST). We pre-train fPaSST on the weakly annotated AudioSet using Knowledge Distillation [27], obtaining a mAP of 0.484.

BEATs: Likewise, BEATs [10] is also based on the AST [26] architecture; it was trained in an iterative, self-supervised procedure on AudioSet, where the BEATs encoder and tokenizer were alternately updated. In our experiments, we rely on the checkpoint of BEATs after the third iteration, where both the tokenizer and the encoder were fine-tuned on the weak labels of AudioSet.

4. TRAINING PIPELINE

In this section, we describe the pre-training routine on AudioSet strong and how the pre-trained models are fine-tuned on the Task 4 datasets in the proposed multi-iteration, multi-stage training procedure. An overview of the full training pipeline is shown in Figure 2. The full system architecture, depicted in Figure 1, is involved in all iterations and stages of Figure 2. The pre-trained transformers
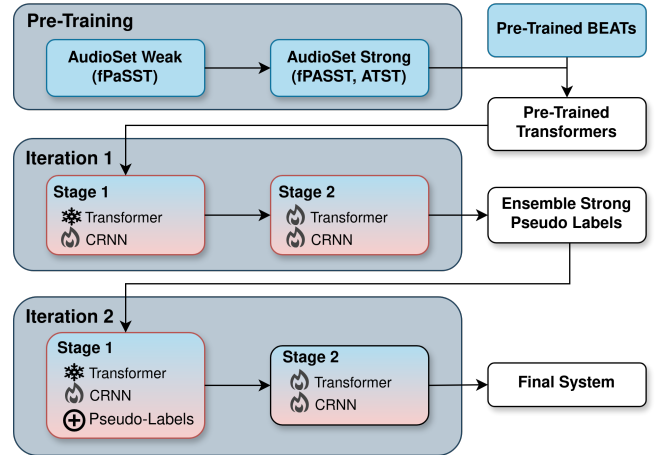

Figure 2: Training Pipeline. The snow flake symbol denotes frozen parameters, the flame that a model is trained in a particular stage.

(blue block in Figure 1) are used as frozen audio embedding models in Stage 1 and fine-tuned together with the CRNN (red blocks in Figure 1) in Stage 2. The pseudo-labels are generated from an ensemble after Iteration 1 and used as additional prediction targets in Stage 1 of Iteration 2. In the following, we abbreviate Iteration {1,2} and Stage {1,2} as *I{1,2}* and *S{1,2}*, respectively.

4.1. Pre-Training on AudioSet strong

We hypothesize that the transformer models would benefit from additional pre-training on a large dataset strongly annotated for various acoustic events. To this end, we add a BiGRU block with 1024 units that processes the output of the transformer. We pre-train for 10 epochs on AudioSet strong [28], a subset of AudioSet that holds around 86,000 strongly labeled examples with annotations for 456 event classes. While ATST and, in particular, fPaSST benefit from this pre-training, there was no effect on the downstream task performance of BEATs; therefore, we only pre-train ATST and fPaSST on AudioSet strong. We select the checkpoint with the highest PSDS1 score on the AudioSet strong validation set for downstream training.

4.2. Multi-Stage Training

Inspired by [16], *I1* and *I2* are both split into two training stages. In *S1*, the CRNN (CNN + BiGRU) is trained from scratch while the large transformer model is frozen. This setup corresponds to the training of the baseline system with slightly different hyperparameters and additional data augmentations (as shown in Table 1).

In *S2*, the CRNN is initialized with pre-trained weights from *S1*, and both the CRNN and the transformer model are fine-tuned. As the system already performs well in its initial state, the transformer can rely on a high-quality self-supervised loss computed on the larger unlabeled set. Aligned with [16], in *S2*, we compute the interpolation consistency (ICT) loss [29] in addition to the mean teacher (MT) loss [30]. In both stages, we choose the best model based on the validation set by computing the sum of PSDS1 on the strongly labeled synthetic data, PSDS1 on external strongly labeled real data, and mpAUC on the MEASTRO validation set.

4.3. Multi-Iteration Training

After completing *I1*, we build an ensemble (see *Ensemble Stage 2* in Table 2) of multiple ATST, fPaSST, and BEATs models. This ensemble is used to compute strong pseudo-labels for all audio clips in the training set by averaging the frame-wise logits of the individual

Aug. Method	Target	HP	Pipeline
DIR [20]	All	p=0.5	$I\{1,2\}.S2$
Wavmix [33]	Str.	p=0.5,α=0.2	$I\{1,2\}.S\{1,2\}$
Freq-MixStyle [18]	All	p=0.5,α=0.3	$I1.S\{1,2\},I2.S2$
Mixup [33]	All	p=0.5,α=0.2	$I\{1,2\}.S\{1,2\}$
Time-Masking	DES. Str.	s=[0.05,0.3]	$I\{1,2\}.S2$
FilterAugment [17]	All	linear,p=0.8	$I1.S\{1,2\},I2.S2$
Freq-Warping [8]	All	p=0.5	$I\{1,2\}.S2$

Table 1: The table lists data augmentation methods, the data subset they are applied to (**Target**), hyperparameters (**HP**), and the respective iteration and stage they are used in (**Pipeline**). p is the probability for applying the augmentation method; α parameterizes Beta distributions; s specifies the masking ratio interval; and (*DES.*) *Str.* refers to strongly annotated audio clips (from DESED).

models. In *S1* of *I2*, we then use the pseudo-labels as additional prediction targets. We found that BCE is superior to MSE for computing the pseudo-label loss, and interestingly, using the pseudo-label loss only improves performance in *S1* of *I2* (see Table 4).

5. EXPERIMENTAL SETUP

5.1. Audio Pre-processing and Augmentation

For all models, we convert audio clips to 10 seconds in length at a 16 kHz sampling rate. For the CNN, we match the baseline settings and compute Mel spectrograms with 128 Mel bins using a window length of 128 ms and hop size of 16 ms. For the transformers, we use their original feature extraction pipelines [9, 10, 25].

Table 1 details all the data augmentation methods used in our training pipeline. In contrast to the baseline, we apply Cross-Dataset Mixup and Cross-Dataset Freq-MixStyle. That is, we mix audio clips from MAESTRO and DESED instead of keeping them separate. In the case of Mixup, we allow the loss to be calculated for all partially active classes, irrespective of the audio clip's dataset origin (see Section 5.3). For Wavmix and Mixup, we mix the pseudo-labels accordingly.

5.2. Datasets and Optimization

We use the DESED [7] and MAESTRO [4] datasets as provided for Task 4 in the DCASE 2024 challenge and, additionally, approximately 7,000 strongly annotated clips extracted from AudioSet strong according to [31]. We refer the reader to [24] for a detailed description of the data setup.

The training data can be seen as the union of five subsets: MAESTRO strong and DESED: real strong, synthetic strong, weakly annotated, and unlabeled. We draw batches of (12, 10, 10, 20, 20) and (56, 40, 40, 72, 72) samples from these datasets in *S1* and *S2*, respectively. The model is trained to minimize BCE loss on all (pseudo-)labeled audio clips and MSE loss for the self-supervised MT [30] and ICT [29] methods. We compute a weighted sum of all losses and tune the individual weights for all iterations and stages. AdamW [32] with weight decays of 1e-2 and 1e-3 is used in *S1* and *S2*, respectively. Learning rates are listed in Table 2.

5.3. Handling Heterogeneous Sound Event Classes

The DESED and MAESTRO datasets are annotated with two different sets of sound event classes. We adopt the baseline [13] strategy, in which the loss for an audio clip is calculated only on the dataset-specific event classes and mapped event classes, as explained in

the following: To exploit the fact that the DESED and MAESTRO classes are not fully disjoint but partly represent the same concepts, the baseline introduces class mappings. For example, when the classes *people talking*, *children voices*, or *announcement* are active in a MAESTRO clip, the corresponding DESED class *Speech* is set to the same confidence value. In addition, we also include a mapping from MAESTRO to DESED classes. Specifically, we set the values of the MAESTRO classes *cutlery and dishes* and *people talking* to 1 if the DESED classes *Dishes* and *Speech* are present. This is also performed for weak class labels.

5.4. Postprocessing

For model selection and hyperparameter tuning, we stick with the same class-wise median filter used in the baseline system [13]. After selecting the best configurations for each model, we apply the recently introduced Sound Event Bounding Boxes (SEBBs) [34] method for postprocessing. We use class-wise parameters and tune them by using linearly spaced search grids (8 values) for step filter length (0.38 to 0.66), relative merge threshold (1.5 to 3.25), and absolute merge threshold (0.15 to 0.325).

6. RESULTS

In this section, we present the results of the described models (Section 3) in the introduced training pipeline (Section 4). Table 2 lists the best configuration and the corresponding results on the test set for each architecture in both iterations and stages. The table lists the sequence pooling method (**Seq.**) and the CNN (**lr_cnn**), RNN (**lr_rnn**), and Transformer (**lr_tf**) learning rates. **lr_dec** indicates the layer-wise learning rate decay for the transformers as used in [16].

In *I1.S1*, in which the transformers are frozen, BEATs seems to extract the embeddings of the highest quality, followed by fPaSST and ATST. *I1.S1* with BEATs is very similar to the baseline [13] and achieves a similar rank score with a slight performance increase in our setup. Compared to *I1.S1*, all three transformers demonstrate a large increase in rank score when fine-tuned on the Task 4 datasets in *I1.S2*. Notably, the three transformers have different strengths, with ATST and BEATs achieving the best scores on MAESTRO and DESED clips, respectively. *Ensemble Stage 2* denotes an ensemble of 46 models resulting from *I1.S2*, including ATST, fPaSST, and BEATs trained in different configurations. We use *Ensemble Stage 2* to generate strong pseudo-labels for all audio clips in the dataset.

The additional pseudo-label loss in *I2.S1* boosts performance substantially, with all three transformers achieving a higher rank score compared to *I1.S2*. The top rank scores for all models are achieved in *I2.S2*, with ATST obtaining the highest rank score.

Table 3 presents the top configurations of ATST, fPaSST, and BEATs from *I2.S2* with the state-of-the-art postprocessing method cSEBBs [34] applied. *ATST* and *ATST DT*, a variant of ATST that is trained on all available audio clips included in the Task 4 development set, were submitted as single models to the challenge. *ATST DT* using cSEBBs postprocessing achieves a PSDS1 of 0.692 on the public evaluation set of DESED, improving over the previous state of the art (0.686 PSDS1) [34].

6.1. Ablation Study

Table 4 shows the results of ATST for *I2.S1* and *I2.S2* trained in different configurations to analyze the design choices related to the heterogeneous datasets and the pseudo-label loss. For settings - *DESED* and - *MAESTRO*, the proposed system is trained only on MAESTRO and DESED data, respectively. We find that training on

		Model	lr_cnn	lr_rnn	lr_tf	lr_dec	Seq.	mpAUC	PSDS1	Rank Score
Iteration 1	Stage 1	ATST	1e-3	1e-3	-	-	int. lin.	0.702 ± 0.008	0.493 ± 0.012	1.195 ± 0.012
		fPaSST	1e-3	1e-3	-	-	int. nearest	0.709 ± 0.021	0.502 ± 0.010	1.212 ± 0.027
		BEATs	1e-3	1e-3	-	-	int. nearest	0.719 ± 0.004	0.509 ± 0.003	$\mathbf{1.228 \pm 0.006}$
	Stage 2	ATST	1e-4	1e-3	1e-4	0.5	int. nearest	0.739 ± 0.017	0.520 ± 0.005	$\mathbf{1.259} \pm 0.020$
		fPaSST	1e-4	1e-3	1e-4	1	int. nearest	0.726 ± 0.021	0.514 ± 0.008	1.24 ± 0.027
		BEATs	1e-4	1e-3	1e-4	1	int. lin.	0.713 ± 0.002	0.539 ± 0.004	1.252 ± 0.003
	Ensemble Stage 2		-	-	-	-	mix	0.735	0.569	1.303
Iteration 2	Stage 1	ATST	5e-4	5e-4	-	-	avg. pool	0.741 ± 0.017	0.536 ± 0.006	$\mathbf{1.277 \pm 0.012}$
		fPaSST	5e-4	5e-4	-	-	int. nearest	0.722 ± 0.011	0.526 ± 0.004	1.248 ± 0.012
		BEATs	5e-4	5e-4	-	-	int. nearest	0.724 ± 0.011	0.537 ± 0.005	1.262 ± 0.010
	Stage 2	ATST	1e-5	1e-4	1e-4	0.5	avg. pool	0.750 ± 0.004	0.548 ± 0.004	$\mathbf{1.298 \pm 0.006}$
		fPaSST	5e-5	5e-4	1e-4	1	int. nearest	0.719 ± 0.013	0.539 ± 0.003	1.259 ± 0.015
		BEATs	5e-5	5e-4	1e-4	1	int. nearest	0.729 ± 0.005	0.557 ± 0.005	1.286 ± 0.009

Table 2: The table presents the results of ATST, fPaSST, and BEATs for both iterations and stages on the official development test set. For each model, we list the best configuration in terms of the sequence length adaptation method (**Seq.**), where *int. lin.*, *int. nearest*, *avg. pool*, and *mix* denote linear and nearest-exact interpolation, adaptive average pooling, and a mixture of these methods, respectively. *Ensemble Stage 2* is used to generate the pseudo-labels for Iteration 2. **Rank Score** denotes the sum of **mpAUC** and **PSDS1**.

Model	mpAUC	PSDS1 MF	PSDS1*	Ev. PSDS1*
ATST	0.750	0.548	0.617	0.684
fPaSST	0.719	0.539	0.601	0.681
BEATs	0.729	0.557	0.622	0.683
ATST DT	✗	✗	✗	0.692

Table 3: Results for best single-model configurations of ATST, fPaSST, and BEATs from *I2.S2*. **PSDS1** lists results with a median filter; **PSDS1*** results using cSEBBs postprocessing [34]; and **Ev. PSDS1*** lists results on the DESED public evaluation set with cSEBBs postprocessing. *ATST DT* denotes the best ATST configuration trained on the full development set.

DESED and MAESTRO simultaneously is beneficial for the performance on both datasets, which coincides with the finding reported for the baseline system [13]. For both stages of *I2*, excluding MAESTRO clips when calculating the self-supervised losses (- *SSL MAESTRO*) and not mapping event classes from MAESTRO to DESED (- *MAESTRO-DESED Map.*, see Section 5.3) leads to a performance decrease. However, we find no clear answer to the question of whether the SSL loss should be calculated on all classes or only on the dataset-specific classes of an audio clip (+/- *SSL class mask*); *S1* and *S2* benefit from different settings. Interestingly, using the pseudo-label loss in *I2.S2* (+ *Pseudo Loss*) does not increase the rank score. Therefore, the setup in *I1.S2* and *I2.S2* remains identical, which demonstrates that a well-trained CRNN from *S1* can have a large impact on the performance achieved in *S2*. We also tried to use separate heads for predictions on DESED and MAESTRO classes and realized this with an additional single BiGRU layer per dataset (+ *Separate RNN layer*), which resulted in a performance decrease. Further obvious choices, such as thresholding the pseudo-labels by 0.5 (+ *Hard Pseudo*) and calculating the pseudo-label loss on all classes (+ *Pseudo All Classes*) instead of only dataset-specific classes, are inferior to our proposed strategy as well.

7. CONCLUSION

This paper presented a multi-iteration, multi-stage training routine for fine-tuning transformers on the SED task with heterogeneous datasets. We showed that the performance of all tested systems

System	mpAUC	PSDS1	Rank Score
ATST *I2.S1*	0.741	0.536	**1.277**
- DESED	0.724	-	-
- MAESTRO	-	0.531	-
- SSL MAESTRO	0.741	0.535	1.276
- MAESTRO-DESED Map.	0.717	0.530	1.247
+ SSL class mask	0.740	0.530	1.27
+ Separate RNN layer	0.714	0.531	1.244
+ Hard Pseudo	0.706	0.538	1.244
+ Pseudo All Classes	0.717	0.534	1.25
ATST *I2.S2*	0.750	0.548	**1.298**
- SSL MAESTRO	0.743	0.546	1.289
- MAESTRO-DESED Map.	0.749	0.547	1.297
- SSL class mask	0.749	0.544	1.293
+ Pseudo Loss	0.746	0.552	1.297

Table 4: Ablation Study on design choices related to the heterogeneous datasets and the pseudo-label loss used in *I2.S1*. The study is performed on the top single model, ATST, trained in *I2.S1* (upper part) and in *I2.S2* (lower part).

monotonously increases throughout both iterations and stages. The proposed method led to a new state-of-the-art performance of 0.692 in PSDS1 on the DESED public evaluation set and achieved the top rank in Task 4 of the DCASE 2024 challenge. We specifically studied design choices related to the heterogeneous datasets and found that simultaneously training on DESED and MAESTRO leads to a performance increase on both datasets compared to training the system on a single dataset.

8. ACKNOWLEDGMENT

9. REFERENCES

[1] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *WASPAA*, 2005.

[2] C. Debes, A. Merentitis, S. Sukhanov, M. E. Niessen, N. Frangiadakis, and A. Bauer, "Monitoring activities of daily living in smart homes: Understanding human behavior," *IEEE Signal Process. Mag.*, 2016.

[3] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound - proof of concept on human mimicking doll falls," *IEEE Trans. Biomed. Eng.*, 2009.

[4] I. Martín-Morató and A. Mesaros, "Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2023.

[5] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2020.

[6] S. Park, A. Bellur, D. K. Han, and M. Elhilali, "Self-training for sound event detection in audio mixtures," in *ICASSP*, 2021.

[7] N. Turpault, R. Serizel, A. Parag Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *DCASE Workshop*, 2019.

[8] X. Li and X. Li, "ATST: audio representation learning with teacher-student transformer," in *Interspeech*, 2022.

[9] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Interspeech*, 2022.

[10] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "Beats: Audio pre-training with acoustic tokenizers," in *ICML*, 2023.

[11] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *ICASSP*, 2022.

[12] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," in *ICASSP*, 2020.

[13] S. Cornell, J. Ebbers, C. Douwes, I. Martín-Morató, M. Harju, A. Mesaros, and R. Serizel, "DCASE 2024 task 4: Sound event detection with heterogeneous data and missing labels," *CoRR*, 2024.

[14] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," in *DCASE Workshop*, 2018.

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

[16] N. Shao, X. Li, and X. Li, "Fine-tune the pretrained atst model for sound event detection," in *ICASSP*, 2024.

[17] H. Nam, S. Kim, and Y. Park, "Filteraugment: An acoustic environmental data augmentation method," in *ICASSP*, 2022.

[18] 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.

[19] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *DCASE Workshop*, 2022.

[20] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *EUSIPCO*, 2023.

[21] J. Ebbers and R. Haeb-Umbach, "Pre-training and self-training for sound event detection in domestic environments," DCASE Challenge, Tech. Rep., 2022.

[22] J. W. Kim, S. W. Son, Y. Song, H. K. Kim, I. H. Song, and J. E. Lim, "Semi-supervsied learning-based sound event detection using freuqency dynamic convolution with large kernel attention for DCASE challenge 2023 Task 4," DCASE Challenge, Tech. Rep., 2023.

[23] 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.

[24] F. Schmid, P. Primus, T. Morocutti, J. Greif, and G. Widmer, "Improving audio spectrogram transformers for sound event detection through multi-stage training," DCASE Challenge, Tech. Rep., 2024.

[25] X. Li, N. Shao, and X. Li, "Self-supervised audio teacher-student transformer for both clip-level and frame-level tasks," *IEEE ACM Trans. Audio Speech Lang. Process.*, 2024.

[26] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," in *Interspeech*, 2021.

[27] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *ICASSP*, 2023.

[28] S. Hershey, D. P. W. Ellis, E. Fonseca, A. Jansen, C. Liu, R. C. Moore, and M. Plakal, "The benefit of temporally-strong labels in audio event classification," in *ICASSP*, 2021.

[29] X. Zheng, Y. Song, I. McLoughlin, L. Liu, and L. Dai, "An improved mean teacher based method for large scale weakly labeled semi-supervised sound event detection," in *ICASSP*, 2021.

[30] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *ICLR Workshop*, 2017.

[31] S. Xiao, J. Shen, A. Hu, X. Zhang, P. Zhang, and P. Yan, "Sound event detection with weak prediction for dcase 2023 challenge task4a," DCASE Challenge, Tech. Rep., 2023.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[33] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[34] J. Ebbers, F. G. Germain, G. Wichern, and J. L. Roux, "Sound event bounding boxes," *accepted at Interspeech*, 2024.