IMPROVING AUDIO SPECTROGRAM TRANSFORMERS FOR SOUND EVENT DETECTION THROUGH MULTI-STAGE TRAINING

Technical Report

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

This technical report describes the CP-JKU team's submission for Task 4 *Sound Event Detection with Heterogeneous Training Datasets and Potentially Missing Labels* of the DCASE 24 Challenge. We fine-tune three large Audio Spectrogram Transformers, PaSST, BEATs, and ATST, on the joint DESED and MAESTRO datasets in a two-stage training procedure. The first stage closely matches the baseline system setup and trains a CRNN model while keeping the large 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 all three 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, boosting single-model performance substantially. Additionally, we pre-train PaSST and ATST on the subset of AudioSet that comes with strong temporal labels, before fine-tuning them on the Task 4 datasets[1].

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

1. INTRODUCTION

The task of Sound Event Detection (SED) is to recognize and classify specific sound events in audio signals, including the temporal location of the events. Developing reliable SED systems allows their use in important real-world applications, such as security and surveillance [1], smart homes [2], or health monitoring [3]. A main driver of research in this field is the annual DCASE Challenge, with Task 4 specifically tackling Sound Event Detection. This technical report describes the CP-JKU team's submission to DCASE Challenge 2024 Task 4: *Sound Event Detection with Heterogeneous Training Datasets and Potentially Missing Labels* [4].

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 acute lack of strongly annotated sound event datasets [5]. Hence, previous editions of Task 4 focused on learning from weakly labeled data [6], semi-supervised learning strategies [7], and utilizing synthetic strongly labeled data [8] in an attempt to develop systems that perform well on real-world strongly labeled sound clips. While Task 4 has been based on the DESED dataset [8] in previous years, the key novelty of the 2024 edition is a unified setup including a second dataset, MAESTRO Real [5]. 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 performance on both datasets, as demonstrated for the baseline system [4].

The main contributions of this work can be summarized in the following points: **(1)** We demonstrate that multiple different pre-trained transformer models (ATST [9], PaSST [10], and BEATs [11]) can be fine-tuned on the Task 4 datasets to achieve high performance. **(2)** Pre-training on the temporally-strong annotated portion of AudioSet [12] (AudioSet strong) can improve performance for the frame-wise pre-trained model ATST and is necessary for the clip-wise pre-trained PaSST to obtain high-quality frame-wise predictions. **(3)** Combining fine-tuned ATST, PaSST, and BEATs models leads to a diverse ensemble that can be used to create high-quality pseudo-labels. **(4)** Using the computed pseudo-labels in a second training iteration dramatically improves single-model performance, leading to a relative increase of 25.6% in terms of polyphonic sound detection score [13, 14] (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. On DESED, we achieve a new state-of-the-art performance on the public evaluation set, increasing the single-system performance from .686 [13] to .692 in terms of PSDS1.

2. DATASETS

The development set is composed of two datasets: DESED [8] and MAESTRO Real [5]. For common processing, all audio in the training set is converted to clips of 10 seconds in length. For the MAESTRO dataset, we strictly follow the train-test-validation split established by the baseline system [4]. As for DESED, we use the following subsets:

- Weakly labeled: clip-wise labels, 1,267 / 158 for train. / valid.
- Unlabeled: 13,057 unlabeled clips
- Synthetic strong: 10,000 / 2,500 strongly labeled synthetic clips for train. / valid.
- AudioSet strong: 3,435 strongly labeled real clips
- External strong: 6,426 / 957 additional strongly labeled real clips for train. / valid. from AudioSet strong as used in [15]
- Test: 1,168 strongly labeled real clips as in baseline setup [4]

[1]Code: `https://github.com/CPJKU/cpjku_dcase24`

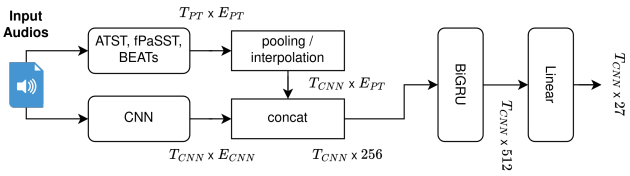

Figure 1: Overview of System Architecture.

We noticed that the provided AudioSet strong subset overlaps with weakly labeled, unlabeled, and test sets. Therefore, we remove 1,355 files from the unlabeled set and 153 files from the weakly labeled set to avoid oversampling individual audio clips. Additionally, we remove the 35 files that overlap with the test set from the AudioSet strong set to obtain a more accurate estimate of the generalization performance.

2.1. Cross mapping sound event classes

To exploit the fact that the DESED and MAESTRO classes are not fully disjoint but partly represent the same concepts, the baseline system 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 classes to DESED clips. 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.

3. SYSTEM ARCHITECTURE

Figure 1 depicts an overview of our SED system. The system is very similar to the baseline [4]. However, besides BEATs [11], we experiment with two additional Audio Spectrogram Transformers, ATST [9] and PaSST [10]. Section 3.2 introduces modifications to the PaSST architecture for allowing high-quality frame-wise predictions, and in Section 4.1, we describe pre-training of PaSST and ATST on AudioSet strong. In addition to adaptive average pooling, we experiment with linear and nearest-exact interpolation to align transformer and CNN sequence lengths. The BiGRU block consists of two bidirectional GRU layers with a dimension of 256.

3.1. ATST-Frame

ATST-Frame [16](denoted only ATST in the following) was specifically designed to produce a sequence of frame-level audio embeddings instead of a single global clip-level representation and is thus particularly suited for SED. The architecture of ATST is based on that of the Audio Spectrogram Transformer (AST) [17] and it is trained in a self-supervised manner via masked spectrogram modeling in a student-teacher scheme on AudioSet. In our experiments, we use a checkpoint of ATST that is further fine-tuned on the weak labels of AudioSet.

3.2. fPaSST

The Patchout faSt Spectrogram Transformer (PaSST) [10] is an improved version of the original AST [17] that shortens the training time and improves the performance via patchout regularization. PaSST uses global classification tokens, which are ideal for tagging and classification tasks but are not designed for inferring the precise temporal occurrence of acoustic events. We thus adopt the architecture of PaSST to return frame-level predictions and call the resulting model Frame-PaSST (fPaSST). fPaSST uses three input convolutions to convert the input spectrogram to a tensor of size $16 \times 128 \times 250$ (channel $\times$ frequency $\times$ time). The result is then converted to 250 768-dimensional tokens via another convolution with kernel size 128×2. We modify the positional encoding accordingly and initiate all parameters except the first three convolutions with parameters taken from a vision transformer. We pre-trained the resulting model on the weakly annotated AudioSet using Knowledge Distillation as described in [18], obtaining a mAP of 0.484.

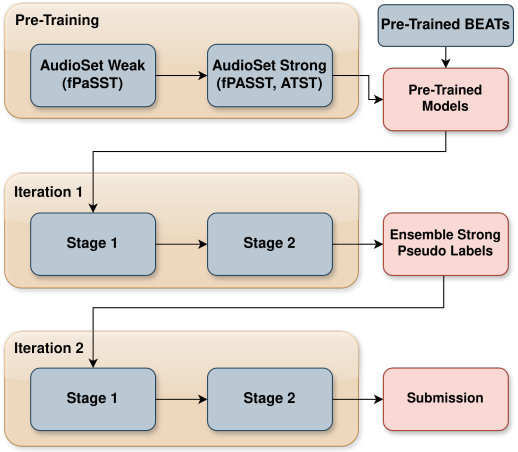

Figure 2: Overview of Training Pipeline.

3.3. BEATs

Likewise, BEATs [11] is also based on the AST [17] architecture; it takes rectangular spectrogram patches as input and returns one embedding vector for each, making it suitable for SED. The model was trained in an iterative, self-supervised procedure where the BEATs encoder learned representations from a frozen audio-tokenizer model that was itself learned from the BEATs encoder after every iteration. 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 a multi-iteration, multi-stage training procedure. An overview of the full training pipeline is shown in Figure 2. The multi-stage setup closely follows the strategy outlined in [19], and a multi-iteration setup with learning from pseudo-labels showed to improve performance drastically in [20] and [21]. 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 audio embedding 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 fPaSST or ATST. We

train both models for 10 epochs on AudioSet strong [22], a subset of AudioSet that holds around 86.000 strongly labeled examples with annotations for 456 event classes. The learning rate for the pre-trained ATST and PaSST encoders is linearly increased from 0 to 4e-5, while the learning rate for the uninitialized BiGRU block is linearly decayed from 1e-3 to 4e-5 in the first four epochs. We select the checkpoint with the highest PSDS1 score on the AudioSet strong validation set for Task 4 downstream training.

4.2. Multi-Stage Training

I1 and I2 are both split into two training stages. In S1, the CNN and BiGRU are trained from scratch while the large transformer model is kept frozen. This setup corresponds to the training of the baseline system with slightly different hyperparameters and additional data augmentations.

In S2, the CNN and BiGRU are initialized with pre-trained weights from S1, and the transformer model is fine-tuned. As the system already performs well in its initial state, the transformer can rely on high-quality self-supervised loss computed on the larger unlabeled set. Aligned with [19], in S2, we compute interpolation consistency loss [23] in addition to the mean teacher loss.

In both stages, we choose the best model based on the validation metrics of the student. Specifically, we compute 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 models. In S1 of I2, we then compute BCE between the model's predictions and the pseudo-labels as an additional loss term. We found that BCE is superior to MSE, and interestingly, using the pseudo-label loss only helps in S1 of I2. We hypothesize that the CRNN picks up relevant information from the pseudo-labels in S1 and transfers it to the transformer model via the high self-supervised loss weights in S2 of I2.

5. EXPERIMENTAL SETUP

5.1. Audio Pre-processing

For all models, we resample audio clips at 16 kHz. For the CNN, we match the baseline settings and compute mel spectrograms with 128 mel bins using a window length of 128 ms and a hop size of 16 ms. For ATST, fPaSST, and BEATs, we match the pre-training setup and compute mel spectrograms with 64, 128, and 128 mel bins, respectively. All transformers use a hop size of 10 ms; the window size of fPaSST and BEATs is set to 25 ms; and for ATST, it is 64 ms.

5.2. Data Augmentation

Table 1 presents in detail all the data augmentation methods we use 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 modify the class mask and allow the loss to be calculated for all active classes, irrespective of the audio clip's dataset origin. For Wavmix and Mixup, we mix the pseudo-labels accordingly.

Aug. Method	**Target**	**HP**	**Pipeline**
DIR [24]	All	p=0.5	I{1,2}.S2
Wavmix [25]	Str.	p=0.5,α=0.2	I{1,2}.S{1,2}
Freq-MixStyle [26]	All	p=0.5,α=0.3	I1.S{1,2},I2.S2
Mixup [25]	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 [27]	All	linear,p=0.8	I1.S{1,2},I2.S2
Freq-Warping [9]	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; and *Str.* refers to strongly annotated audio clips.

5.3. Data Sampling and Optimization

We summarize the training data in five subsets: MAESTRO, DESED real strong, synth. strong, weakly annotated, and unlabeled. In S1, we draw batches of (12, 10, 10, 20, 20) samples, and in S2, we draw batches of (56, 40, 40, 72, 72) samples from these datasets. The model needs to optimize six losses in parallel: MAESTRO strong, DESED real strong, synth. strong, weak, self-supervised loss, and pseudo-label loss. Besides the MSE loss used for the self-supervised loss, the BCE loss is computed for all others. We compute a weighted sum of all losses and tune the individual weights for all iterations and stages. In contrast to the baseline, we also compute the self-supervised loss on MAESTRO clips.

We use the AdamW [28] optimizer with weight decays of 1e-2 and 1e-3 in S1 and S2, respectively. Learning rates are listed in Table 2.

5.4. Postprocessing

For model selection and hyperparameter tuning, we stick with the class-wise median filter used in the baseline system [4]. After selecting models for submission, we apply the recently introduced Sound Event Bounding Boxes [29] method for post-processing. We use class-wise parameters and obtain 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). We follow the strategy of the baseline [4] and tune these hyperparameters on the development-test set, as the class-wise median filter lengths of the baseline system are tuned on the development-test set as well.

6. RESULTS

In this section, we present the results of the described models (Section 3) in the introduced training pipeline (Section 4). In Section 6.1, systems selected for submission are presented.

Table 2 lists the results for the best configuration of each model in terms of sequence length adaptation method (**Seq.**) and learning rate in each iteration and stage. Furthermore, the CNN (**lr_cnn**), RNN (**lr_rnn**), and Transformer (**lr_tf**) learning rates are listed. **lr_dec** describes layer-wise learning rate decay for the Transformer models as used in [19].

In I1.S1, in which the transformer models 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

		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	0.75 ± 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	0.7286 ± 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. For each model, we list the best configuration in terms of the sequence length adaptation method (**Seq.**). *Ensemble Stage 2* is used to generate the pseudo-labels for Iteration 2. **Rank Score** denotes the sum of **mpAUC** and **PSDS1**.

ID	Models	#	Dev-Test	mpAUC	PSDS1 MF	PSDS1 SEBB	Eval PSDS1 SEBB
S1	ATST I2.S2	1	✗	0.749	0.548	0.617	0.684
S2	ATST I2.S2	1	✓	-	-	-	0.692
S3	Ensemble I2.S1 + I2.S2	18	✗	0.743	0.569	0.632	0.721
S4	Ensemble I2.S1 + I2.S2	15	✓	-	-	-	0.729

Table 3: Final Submissions: **#** lists the number of models; the flag **Dev-Test** indicates that we use the full development set for training; **PSDS1 MF** lists results with a median filter; **PSDS1 SEBB** lists DESED test set results using SEBB postprocessing [29]; and **Eval PSDS1 SEBB** lists results on the public evaluation set with SEBB postprocessing.

setup [4] and achieves a similar rank score with a slight performance increase in our setup.

Compared to the rank scores in I1.S1, in I1.S2, all three transformers demonstrate a large increase in rank score when fine-tuned on the Task 4 datasets. Notably, the three transformer models have different strengths, with ATST achieving the best score on MAESTRO clips while BEATs obtains the best score on DESED clips. *Ensemble Stage 2* denotes an ensemble of 46 models resulting from I1.S2, including ATST, fPaSST, and BEATs trained in different configurations. While the performance in terms of PSDS1 benefits largely from ensembling a large number of models, the mpAUC is even slightly worse compared to the best single model after I1.S2 (ATST). 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 higher performance in I2.S1 in terms of rank score compared to I1.S2. Interestingly, ATST, which achieves the lowest performance in I1.S1, has the highest performance in I2.S1 outperforming the other models in particular on the MAESTRO clips.

The top rank scores for all models are achieved in I2.S2, with ATST obtaining the best single-model performance. Notably, the pseudo-label loss is not used in I2.S2, as it does not increase the rank score, demonstrating that a well-trained CRNN from S1 is instrumental for high performance in S2.

6.1. Final Submissions

For the final submissions shown in Table 3, we select the top single-model, ATST, after I2.S2, and build an ensemble consisting of ATST, fPaSST, and BEATs models obtained in I2. We repeat the full training process for the two selections and include the test data of MAESTRO and DESED for training to make use of the full development set. In this case, model selection still relies on validation metrics.

As described in Section 5.4, we use SEBBs [29] instead of a class-wise median filter for postprocessing the predictions of all submissions. The resulting performance is listed in Table 3. **PSDS1 MF** and **PSDS1 SEBB** denote the performance on the DESED test set with median filter and SEBB, respectively. Although the median filter lengths of the baseline system are also tuned on the test set, we note that **PSDS1 SEBB** results should be taken with a grain of salt, as the SEBB hyparameters are tuned on the test set. We therefore also report the results on the unseen DESED public evaluation set (**Eval PSDS1 SEBB**). Notably, our best single model (**S2**) improves the state-of-the-art PSDS1 score on the public evaluation set from 0.686 [29] to 0.692. The submitted ensembles (**S3** & **S4**) clearly improve over the single models (**S1** & **S2**) in terms of PSDS1, but interestingly, mpAUC cannot be improved by ensembling.

7. ACKNOWLEDGMENT

The computational results presented were achieved in part using the Vienna Scientific Cluster (VSC) and the Linz Institute of Technology (LIT) AI Lab Cluster. The LIT AI Lab is supported by the Federal State of Upper Austria. Gerhard Widmer's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No 101019375 (Whither Music?).

8. REFERENCES

[1] R. Radhakrishnan, A. Divakaran, and A. Smaragdis, "Audio analysis for surveillance applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 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] 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.

[5] 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.

[6] 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.

[7] S. Park, A. Bellur, D. K. Han, and M. Elhilali, "Self-training for sound event detection in audio mixtures," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

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

[9] X. Li and X. Li, "ATST: audio representation learning with teacher-student transformer," in *23rd Annual Conference of the International Speech Communication Association*, 2022.

[10] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Annual Conference of the International Speech Communication Association*, 2022.

[11] 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 *International Conference on Machine Learning*, 2023.

[12] 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 *International Conference on Acoustics, Speech and Signal Processing*, 2017.

[13] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *International Conference on Acoustics, Speech and Signal Processing*, 2022.

[14] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," in *International Conference on Acoustics, Speech and Signal Processing*, 2020.

[15] 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.

[16] 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.

[17] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," in *Annual Conference of the International Speech Communication Association*, 2021.

[18] F. Schmid, K. Koutini, and G. Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *International Conference on Acoustics, Speech and Signal Processing*, 2023.

[19] N. Shao, X. Li, and X. Li, "Fine-tune the pretrained atst model for sound event detection," in *International Conference on Acoustics, Speech and Signal Processing*, 2024.

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

[21] 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.

[22] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

[23] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

[24] T. Morocutti, F. Schmid, K. Koutini, and G. Widmer, "Device-robust acoustic scene classification via impulse response augmentation," in *European Signal Processing Conference*, 2023.

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

[26] F. Schmid, S. Masoudian, K. Koutini, and G. Widmer, "Knowledge distillation from transformers for low-complexity acoustic scene classification," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2022.

[27] H. Nam, S. Kim, and Y. Park, "Filteraugment: An acoustic environmental data augmentation method," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

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