TRANSFORMER-AIDED AUDIO SOURCE SEPARATION WITH TEMPORAL GUIDANCE AND ITERATIVE REFINEMENT

Technical Report

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

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

ABSTRACT

This technical report presents the CP-JKU team's system for Task 4 *Spatial Semantic Segmentation of Sound Scenes* of the DCASE 2025 Challenge. Building on the two-stage baseline of audio tagging followed by label-conditioned source separation, we introduce several key enhancements. We reframe the tagging stage as a sound event detection task using five Transformers pre-trained on AudioSet Strong, enabling temporally strong conditioning of the separator. We further inject the Transformer's latent representations into a ResUNet separator initialized from AudioSep and extended with a Dual-Path RNN. Additionally, we propose an iterative refinement scheme that reuses the separator's output as input. These improvements raise label prediction accuracy to 73.07% and CA-SDRi to 14.49 for a single-model system on the development test set, substantially surpassing the baseline.

***Index Terms*—** DCASE Challenge, Audio Source Separation, M2D, BEATs, ATST, PaSST, ASIT, ResUNet, AudioSep, time-FiLM, Iterative Refinement

1. INTRODUCTION

Task 4 of the DCASE 2025 Challenge, called *Spatial Semantic Segmentation of Sound Scenes (S5)* [1], challenges participants to develop systems that, given a multi-channel spatial audio recording, detect and isolate the dry sounds of 18 target classes. The task's baseline system [2] adopts a two-stage pipeline. In the first stage, an M2D-based audio tagger [3] detects the presence or absence of target classes in the mixture clip. In the second stage, a class-conditioned ResUNet separator [4] extracts the detected sound sources, where the separator is conditioned via feature-wise linear modulation (FiLM) [5].

In this report, we describe our submission to task 4, which builds upon the baseline system with several key improvements:

- **Sound Event Models of Stage 1:** We distinguish between two types of sound event models, both based on SED Transformers [6]: audio taggers, trained for audio tagging using global labels, and SED models, trained jointly for audio tagging and sound event detection (SED) using precise onsets and offsets. The top-performing ensemble integrates both types of models for detecting target events in the mixture.
- **Transfer Learning from AudioSep:** To improve generalization, we initialize the separation model with pretrained weights from AudioSep [7], a language-conditioned sound separation framework.
- **Time-FiLM:** To improve the separator's conditioning beyond just clip-level class labels, we incorporate a more fine-grained, time-varying signal obtained by adding a trained SED model to stage 2 of the training pipeline. We call this model the *stage 2 SED model*. Notably, the stage 2 SED model is separate from the sound event model used in stage 1.
- **Embedding Injection:** We additionally inject a weighted sum of the intermediate hidden layer representations from the stage 2 SED model directly into the embedding space of the ResUNet.
- **Dual-Path RNN:** To improve long-range dependency modeling, we incorporate a Dual-Path RNN [8] into the ResUNet's embedding space.
- **Iterative Refinement:** We adopt an iterative refinement scheme, where previous separation outputs are fed back into the system to progressively improve separation quality.
- **Additional Training Data:** We collect additional room impulse responses and background noises from the FOA-MEIR data set [9] and additionally train our final submission on the development validation data.

A combination of our best-performing detector and separator achieves a label prediction accuracy of 73.07% and a class-aware signal-to-distortion ratio improvement (CA-SDRi) of 14.48 on the development test split. Our best-performing ensemble further improves these metrics, reaching 77.07% accuracy and a CA-SDRi of 15.04. These results represent a substantial improvement over the baseline system, which achieves 59.80% accuracy and a CA-SDRi of 11.09.

2. TASK SETTING & DATASETS

The goal of the *S5* task is to detect and separate individual sound events from multi-channel time-domain mixtures recorded in realistic environments. Let

$$Y = [y^{(1)}, \ldots, y^{(M)}]^\top \in \mathbb{R}^{M \times T}$$

denote the multi-channel mixture of length T, recorded with M microphones. Each channel $y^{(m)}$ is modeled as:

$$y^{(m)} = \sum_{k=1}^{K} h_k^{(m)} * s_k + n^{(m)} = \sum_{k=1}^{K} x_k^{(m)} + n^{(m)},$$

[*]These authors contributed equally to this work.

where K is the number of active sound sources, s_k is the dry source signal for class c_k, $h_k^{(m)}$ is the room impulse response (RIR) from source k to microphone m, and $n^{(m)}$ is the additive noise.

The objective is to recover the set of individual sources $\{s_1, \ldots, s_K\}$. The source count per mixture can vary from 1 to $K_{max} = 3$, and the number of microphones is $M = 4$.

2.1. DCASE2025 Task 4 Dataset

Each mixture in the development dataset is synthetically generated from four components:

- **Target sound events** s_k: One-shot recordings of the 18 target classes, captured in anechoic conditions.
- **Room impulse responses** $h_k^{(m)}$: Multichannel RIRs recorded in real rooms.
- **Environmental noise** $n^{(m)}$: Environmental background noise.
- **Interference events**: Non-target sounds (i.e., not among the 18 classes).

Mixtures were synthesized at 32kHz/16bit using a modified version of SpatialScaper [10]. Each 10-second clip contains 1–3 target events with SNRs between 5–20 dB, and up to 2 interference events at 0–15 dB. RIRs for all sources in a mixture are drawn from the same microphone position to ensure spatial consistency.

The training and validation sets are generated on the fly using isolated audio sources and metadata, while the test set is pre-synthesized. On-the-fly generation of the training set allows us to access event onsets and offsets, enabling us to train Transformers for SED, as described in Section 3.1 below. Source audio was compiled from both newly recorded material and curated data: target sounds from FSD50K [11] and EARS [12], RIRs and noise from FOA-MEIR [9], and interference events from the Semantic Hearing dataset [13].

2.2. External Datasets

In addition to the provided development set, we incorporated additional RIRs and noise recordings from FOA-MEIR [9]. Specifically, we include the three remaining RIRs from the test subset and 96 RIRs from the Reverb-S subset. Furthermore, we explore the use of 23 additional background noise recordings sourced from the same dataset.

3. SYSTEM ARCHITECTURE

This section details the complete system architecture, as visualized in Figure 1. Section 3.1 describes the SED Transformer architecture, which is used in three ways: 1) as a sound event model in stage 1 to predict events contained in the mixture; 2) as feature extractor in stage 2, injecting supplementary information into the separator's latent space; and 3) as SED model in stage 2 to generate event presence probabilities for temporal guidance of the separator (time-FiLM). Notably, stage 1 and stage 2 use two distinct models. Section 3.2 then explores the separator, its collaboration with the stage 2 SED Transformer, and the iterative separation refinement.

The SED Transformers consume only the first channel of the 4-channel audio mixture (aligned with pre-training on single-channel signals) and outputs predictions for the 18 target classes. We use the same mechanism as the baseline system to convert the predicted probabilities into up to K_{max} one-hot encoded class predictions. The separator is conditioned on these class predictions via FiLM [5].

3.1. SED Transformers

For SED, we start from the AudioSet Strong [14] and AudioSet Weak [15] pre-trained Transformers introduced in [6]. Transformers include ATST-F [16, 17], BEATs [18], fPaSST [19], M2D [20], and ASiT [21], and their corresponding checkpoints are provided via GitHub[1].

These Transformers, denoted as a function g, take a mel spectrogram $\{\mathbf{x}_t\}_{t=1}^{T}$ with T frames as input and produce a sequence of embeddings $\{\hat{\mathbf{z}}_t \in \mathbb{R}^D\}_{t=1}^{S}$ of length S and dimension D:

$$\{\hat{\mathbf{z}}_t\}_{t=1}^{S} = g(\{\mathbf{x}_t\}_{t=1}^{T}) \tag{1}$$

The embeddings are temporally aligned to a resolution of 40 ms (i.e., 250 frames per 10-second audio segment) by adaptive average pooling (for $S > 250$) or linear interpolation (for $S < 250$):

$$\{\hat{\mathbf{e}}_t\}_{t=1}^{250} = \underset{S \to 250}{\text{resample}}\big(\{\hat{\mathbf{z}}_t\}_{t=1}^{S}\big) \tag{2}$$

Frame-wise predictions for the $C = 18$ target classes are then generated by a linear layer (parameterized by $\mathbf{W} \in \mathbb{R}^{C \times D}$ and $\mathbf{b} \in \mathbb{R}^{C}$) followed by a sigmoid activation σ:

$$\{\hat{\mathbf{o}}_t^{(\text{strong})}\}_{t=1}^{250} = \sigma\big(\mathbf{W}\{\hat{\mathbf{e}}_t\}_{t=1}^{250} + \mathbf{b}\big) \tag{3}$$

Corresponding weak predictions $\hat{\mathbf{o}}^{(\text{weak})} \in \mathbb{R}^C$ are obtained by attention-based pooling, as commonly used in SED, for example in the DCASE 2024 baseline for Task 4 [22].

The overall loss is computed as a weighted sum of binary cross-entropy (BCE) losses on strong and weak labels:

$$\begin{aligned}\mathcal{L} = \lambda \cdot \frac{1}{TC} \sum_{t=1}^{T} \sum_{c=1}^{C} \text{BCE}\big(\hat{o}_{t,c}^{(\text{strong})},\ y_{t,c}^{(\text{strong})}\big) \\ + (1-\lambda) \cdot \frac{1}{C} \sum_{c=1}^{C} \text{BCE}\big(\hat{o}_{c}^{(\text{weak})},\ y_{c}^{(\text{weak})}\big)\end{aligned} \tag{4}$$

We either use AudioSet Weak checkpoints in combination with $\lambda = 0$ to train audio tagger, or, we use AudioSet Strong checkpoints in combination with $\lambda = 0.5$ to obtain models capable of also outputting strong predictions (SED Transformers).

3.2. Separation Models

Our separation models rely on the same ResUNet architecture [4] compared to the baseline [2], which converts waveforms to spectrograms, predicts and applies magnitude and phase masks, and then converts the filtered spectrograms back into waveforms using iSTFT. However, instead of training from scratch, we initialize ResUNet with a pre-trained checkpoint from AudioSep [7]. We observe that ResUNet is more compatible with these pre-trained weights than the ResUNetK variant used in the baseline—likely because AudioSep was trained for single-source separation. While AudioSep was originally trained with a hop size of 320, we retain the baseline's hop size of 160 for spectrogram computation, as it consistently yields better performance in our experiments.

The separator consumes all channels of the 4-channel audio mixture $Y \in \mathbb{R}^{4 \times T}$ and computes spectrograms using STFT, resulting in

$$X_{\text{RN}} \in \mathbb{R}^{4 \times F_{\text{RN}} \times T_{\text{RN}}},$$

[1] https://github.com/fschmid56/PretrainedSED

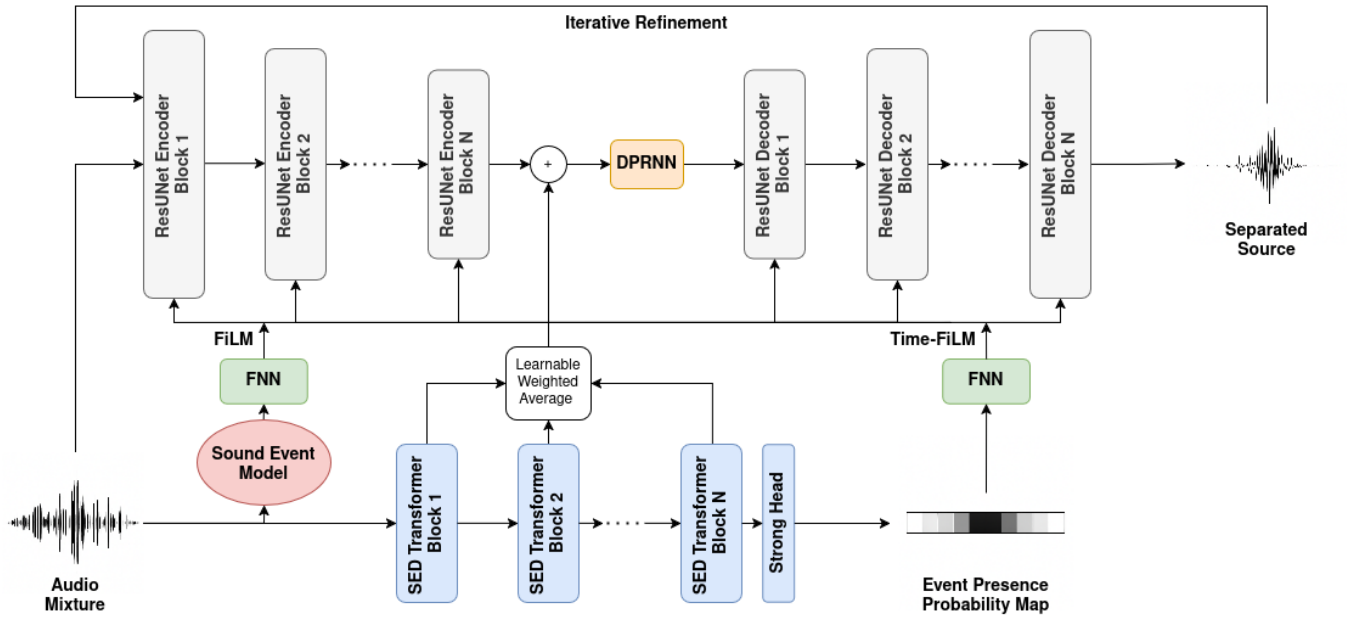

Figure 1: Overview of the proposed system. The sound event model of stage 1 (red) predicts the events contained in the mixture. The stage 2 SED model (blue) calculates an event presence probability map for each class. The predicted events and the event presence probability map are used as conditioning for the separator. Additionally, a learnable weighted average of the features of the stage 2 SED model is injected into the separator's latent space. During the training of the separation model, oracle targets are used instead of the predictions of stage 1.

where F_{RN} is the number of frequency bins and T_{RN} is the number of time frames.

The ResUNet encoder gradually downsamples the spatial dimensions (F_{RN} and T_{RN}) and increases the number of feature channels, producing a latent representation

$$Z_{\text{RN}} \in \mathbb{R}^{C_{\text{RN}} \times F'_{\text{RN}} \times T'_{\text{RN}}},$$

where C_{RN} is the number of latent channels, $F'_{\text{RN}} = F_{\text{RN}}/r_f$, and $T'_{\text{RN}} = T_{\text{RN}}/r_t$. Here, r_f and r_t denote the frequency and time downsampling factors, respectively.

3.2.1. Integration with Stage 2 SED Model

We inject latent representations from the stage 2 SED model into the latent space of the ResUNet. Specifically, we compute a learned weighted sum of the Transformer's encoder outputs, with weights determined during training, and combine it with the features extracted by the ResUNet encoder. This approach, inspired by [23], allows the stage 2 SED model and ResUNet to be trained together for the separation task.

The SED Transformer consumes only the first channel of Y (aligned with pre-training on single-channel signals) and computes a log-mel spectrogram

$$X_{\text{TF}} \in \mathbb{R}^{F_{\text{TF}} \times T_{\text{TF}}},$$

where F_{TF} is the number of mel bins and T_{TF} the number of frames. A patching mechanism is then resulting in:

$$X_{\text{patch}} \in \mathbb{R}^{C_{\text{TF}} \times F'_{\text{TF}} \times T'_{\text{TF}}},$$

where C_{TF} is the Transformer dimension and F'_{TF} and T'_{TF} are the number of patches along frequency and time dimension, respectively. Input to the Transformer blocks is therefore a sequence of length $F'_{\text{TF}} * T'_{\text{TF}}$. After each Transformer block, we reshape the flattened sequence to $F'_{\text{TF}} \times T'_{\text{TF}}$ and obtain a set of latent features from a Transformer with N blocks:

$$\left\{ Z^{(i)}_{\text{TF}} \in \mathbb{R}^{C_{\text{TF}} \times F'_{\text{TF}} \times T'_{\text{TF}}} \right\}_{i=1}^{N}$$

To obtain a unified representation, we learn a scalar score $w_i \in \mathbb{R}$ for each block, apply softmax normalization to get weights α_i, and compute the weighted sum of the features:

$$\alpha_i = \frac{\exp(w_i)}{\sum_{j=1}^{N} \exp(w_j)}, \quad Z_{\text{TF}} = \sum_{i=1}^{N} \alpha_i \cdot Z^{(i)}_{\text{TF}}$$

We apply a linear layer to project C_{TF} to C_{RN} and 2D interpolation over the spatial dimensions ($F'_{\text{TF}} \times T'_{\text{TF}} \rightarrow F'_{\text{RN}} \times T'_{\text{RN}}$) to match the shapes of Z_{TF} and Z_{RN}.

Finally, we combine Z_{TF} and Z_{RN} by element-wise addition.

3.2.2. Dual-Path RNN

To effectively model the global structure within the combined Z_{TF} and Z_{RN} features, we employ a Dual-Path RNN (DPRNN) [8] module, following its successful application in [24]. Specifically, this DPRNN consists of a stack of two identical blocks. Each block processes its input sequentially, first along the time axis with a bidirectional GRU (256 hidden units), and subsequently along the frequency axis with a second BiGRU. The resulting feature map from the DPRNN module is then passed to the ResUNet decoder.

3.2.3. Time-FiLM Conditioning

We introduce *time-FiLM*, a conditioning mechanism that extends FiLM [5] by incorporating a temporal dimension. Instead of using a single, global vector derived from a one-hot label, time-FiLM leverages the time-varying event presence probabilities generated by the stage 2 SED model. For a given target class c, its probability map m_c is selected from $\{\hat{\mathbf{o}}_t^{(\text{strong})}\}_{t=1}^{250}$ and projected into an embedding sequence $e_c \in \mathbb{R}^{250\times E}$ by a dedicated network (FNN in Figure 1). Analogous to FiLM, these embeddings are transformed into time-varying scale and shift parameters. After aligning their temporal resolution to the ResUNet feature maps via interpolation, these parameters modulate the features channel-wise at each time step. This allows the separator to dynamically adapt its behavior based on the instantaneous likelihood of the target event's presence.

3.2.4. Iterative Refinement Scheme

Our final proposal is an iterative refinement scheme for separation, where the separated single-channel source is fed back into ResUNet for refinement. During training, we randomly sample the number of iterations per batch from 1 to N. Gradients are detached and not propagated over multiple iterations. ResUNet's input is extended to five channels (four from the mixture + 1 from the separator source of the previous iteration):

$$X_{\text{RN}} \in \mathbb{R}^{5\times F_{\text{RN}}\times T_{\text{RN}}},$$

In Iteration 1, we input silence for the fifth channel. Interestingly, we find that we can set N=2 during training and extend the maximum number of iterations at inference time to up to 10. This yields small but consistent improvements, with larger gains in the early iterations and diminishing returns in later ones.

4. EXPERIMENTAL SETUP

4.1. SED Transformers

To ensure consistency with the pre-training phase, we resample audio samples to 16 kHz for fine-tuning the SED Transformers [6], matching their original audio pre-processing setup. We fine-tune each SED Transformer for 25,000 steps using a batch size of 32 on a single GPU. We employ a cosine learning rate schedule with 4,000 warm-up steps, along with the Adam optimizer featuring a weight decay between 1×10^{-6} and 1×10^{-5}. The learning rates are set to 4×10^{-4} for M2D and BEATs, 3×10^{-4} for ASiT, 6×10^{-4} for fPaSST, and 1×10^{-4} for ATST-F. Additionally, a layer-wise learning rate decay is applied with a factor between 0.77 and 0.9.

4.2. Separation Models

Following the baseline approach, we train our separation models using audio samples at 32 kHz. We convert the waveforms to spectrograms with a window size of 2048 and a hop size of 160.

We initialize our ResUNet models with the pre-trained AudioSep checkpoint and fine-tune them for 225,000 steps across four GPUs, using a batch size of 4. We set the learning rate to 6×10^{-4} for the pre-trained components and 3×10^{-3} for the newly initialized parts. The training employs a cosine learning rate schedule with 12,000 warm-up steps and an Adam optimizer without weight decay.

To achieve a higher batch size along with faster training, we trained our separation models using 16-bit floating-point precision.

ID	# SED	# Sep	Val	Label Acc ↑	CA-SDRi ↑
S1	108	10	✓	76.87%	14.950
S2	52	6	✓	77.07%	15.042
S3	1	1	✓	73.07%	14.486
S4	30	3	✗	71.73%	14.269

Table 1: Overview of the four systems submitted, with their performance assessed on the test set. Columns *# SED* and *# Sep* denote the number of SED Transformers and separation models in the ensembles for label prediction and separation, respectively. *Val* indicates whether the system was additionally trained on the validation set. *Label Acc* stands for the label prediction accuracy of the SED Transformers and the *CA-SDRi* metric represents the separation quality.

5. RESULTS

Table 1 provides an overview of the four systems we submitted to the challenge. Systems S1, S2, and S4 utilize ensemble methods for stage 1 and stage 2, whereas S3 employs a single M2D tagger model paired with a single ResUNet model. The hyper-parameters for all systems were carefully tuned using the validation set. Following this tuning, S1, S2 and S3 were retrained on a combined dataset that included both the training and validation sets, while S4's ensemble models were trained exclusively on the training set.

System S4, limited to training data alone, achieved a label prediction accuracy of 71.73% and a CA-SDRi of 14.269. In comparison, system S3, despite relying on a simpler single-model approach, surpassed S4 with an accuracy of 73.07% and a CA-SDRi of 14.486. This improvement highlights the advantage of incorporating the validation set into the training process, which likely allowed S3 to perform better on the test set.

Systems S1 and S2, both using ensemble techniques and trained on the combined training and validation sets, demonstrated the strongest performance on the test set. System S1, constructed with 108 sound event models and 10 separation models—totaling 10.8 billion parameters—recorded an accuracy of 76.87% and a CA-SDRi of 14.950. Meanwhile, system S2, derived from a carefully selected subset of S1's models, achieved the highest results with an accuracy of 77.07% and a CA-SDRi of 15.042. However, S2's performance may not generalize as well as S1's on the evaluation set.

6. CONCLUSION

In this technical report, we present our systems submitted to Task 4 of the DCASE 2025 Challenge, significantly enhancing the baseline through three key contributions.

First, we incorporate models trained for Sound Event Detection instead of relying on audio taggers only. This change enables a more precise temporal conditioning of the separator on audio events via time-FiLM. Second, we initialize the ResUNet with pre-trained weights from AudioSep to speed up convergence and boost performance. Third, we advance the separation architecture by integrating latent representations from SED Transformers into the separator's latent space, followed by a Dual-Path RNN to model global time-frequency structures. We also apply iterative refinement during training and inference to further improve results. Finally, we enhance robustness by augmenting the training data with extra Room Impulse Responses (RIRs) and background noise recordings. Together, these innovations deliver significant gains in SED accuracy and separation quality, as evidenced by our systems' strong performance on the challenge test set.

7. ACKNOWLEDGMENT

The computational results presented were obtained 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] M. Yasuda, B. T. Nguyen, N. Harada, R. Serizel, M. Mishra, M. Delcroix, S. Araki, D. Takeuchi, D. Niizumi, Y. Ohishi, *et al.*, "Description and discussion on dcase 2025 challenge task 4: Spatial semantic segmentation of sound scenes," *arXiv preprint arXiv:2506.10676*, 2025.

[2] B. T. Nguyen, M. Yasuda, D. Takeuchi, D. Niizumi, Y. Ohishi, and N. Harada, "Baseline systems and evaluation metrics for spatial semantic segmentation of sound scenes," *arXiv preprint arXiv:2503.22088*, 2025.

[3] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked modeling duo: Towards a universal audio pre-training framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[4] Q. Kong, K. Chen, H. Liu, X. Du, T. Berg-Kirkpatrick, S. Dubnov, and M. D. Plumbley, "Universal source separation with weakly labelled data," *CoRR*, vol. abs/2305.07447, 2023.

[5] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. of the AAAI conference on artificial intelligence*, 2018.

[6] F. Schmid, T. Morocutti, F. Foscarin, J. Schlüter, P. Primus, and G. Widmer, "Effective pre-training of audio transformers for sound event detection," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[7] X. Liu, Q. Kong, Y. Zhao, H. Liu, Y. Yuan, Y. Liu, R. Xia, Y. Wang, M. D. Plumbley, and W. Wang, "Separate anything you describe," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[8] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[9] M. Yasuda, Y. Ohishi, and S. Saito, "Echo-aware adaptation of sound event localization and detection in unknown environments," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[10] I. R. Román, C. Ick, S. Ding, A. S. Roman, B. McFee, and J. P. Bello, "Spatial scaper: A library to simulate and augment soundscapes for sound event localization and detection in realistic rooms," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[11] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2022.

[12] J. Richter, Y. Wu, S. Krenn, S. Welker, B. Lay, S. Watanabe, A. Richard, and T. Gerkmann, "EARS: an anechoic fullband speech dataset benchmarked for speech enhancement and dereverberation," in *Proc. of the Interspeech Conference*, 2024.

[13] B. Veluri, M. Itani, J. Chan, T. Yoshioka, and S. Gollakota, "Semantic hearing: Programming acoustic scenes with binaural hearables," in *Proc. of the ACM Symposium on User Interface Software and Technology, UIST*, 2023.

[14] 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 *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[15] 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 *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[16] X. Li, N. Shao, and X. Li, "Self-supervised audio teacher-student transformer for both clip-level and frame-level tasks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2024.

[17] F. Schmid, P. Primus, T. Morocutti, J. Greif, and G. Widmer, "Multi-iteration multi-stage fine-tuning of transformers for sound event detection with heterogeneous datasets," in *Workshop on Detection and Classification of Acoustic Scenes and Event (DCASE)*, 2024.

[18] 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 *Proc. of the International Conference on Machine Learning (ICML)*, 2023.

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

[20] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, M. Yasuda, S. Tsubaki, and K. Imoto, "M2D-CLAP: masked modeling duo meets CLAP for learning general-purpose audio-language representation," in *Proc. of the Interspeech Conference*, 2024.

[21] S. Atito, M. Awais, W. Wang, M. D. Plumbley, and J. Kittler, "ASiT: Local-global audio spectrogram vision transformer for event classification," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2024.

[22] 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," in *Workshop on Detection and Classification of Acoustic Scenes and Event (DCASE)*, 2024.

[23] C. Hernandez-Olivan, M. Delcroix, T. Ochiai, D. Niizumi, N. Tawara, T. Nakatani, and S. Araki, "Soundbeam meets m2d: Target sound extraction with audio foundation model," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[24] H. Yin, J. Bai, Y. Xiao, H. Wang, S. Zheng, Y. Chen, R. K. Das, C. Deng, and J. Chen, "Exploring text-queried sound event detection with audio source separation," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.