THE MERL SYSTEMS FOR DCASE 2026 CHALLENGE TASK 4

Technical Report

Kohei Saijo[*1], *Yoshiki Masuyama*[*2], *Christoph Boeddeker*[2], *Gordon Wichern*[2], *Julius Richter*[2], *Takahiro Edo*[2], *Jonathan Le Roux*[2]

[1] Information Technology R&D Center, Mitsubishi Electric Corporation, Kanagawa, Japan,
[2] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

ABSTRACT

This technical report describes our spatial semantic segmentation of sound scenes (S5) systems for DCASE 2026 Challenge Task 4. Inspired by the top-ranked system in DCASE 2025 Task 4, we adopt a cascaded framework consisting of universal sound separation (USS) with source counting, source classification, and class-aware refinement. In the first stage, a TF-Locoformer-based USS model separates multi-channel mixtures into single-channel foreground and interference signals. Then, each separated signal is classified into one of 18 foreground classes or as interference. The separated foreground signals are further refined by another TF-Locoformer-based model conditioned on the predicted class labels and the observed mixture. Our best system achieves CA-PI-SDRi of 14.95 dB and mixture accuracy of 78.11% on the dev_test set.

***Index Terms*—** S5, separation, counting, classification

1. INTRODUCTION

This paper describes our system for DCASE 2026 Challenge Task 4, Spatial Semantic Segmentation of Sound Scenes (S5) [1]. This task aims to separate and classify target-class sound sources contained in multichannel sound mixtures. The target classes consist of 18 predefined classes, and the numbers of target and interfering sources in each mixture vary from sample to sample. The evaluation metric is an SNR-based measure, class-aware SDR (CA-SDR), which penalizes errors in estimating class labels and the number of target sources [2]. While the overall task setup largely follows the 2025 version [3], the following two modifications have been introduced to better reflect realistic acoustic scenes: (i) in the 2025 version, a mixture did not contain two or more target sources from the same class, whereas in the 2026 version, multiple target sources from the same class may appear in a mixture; and (ii) accordingly, the evaluation metric, CA-SDR, has been extended to resolve intra-class permutations, resulting in class-aware permutation-invariant SDR (CA-PI-SDR) [4].

In last year's challenge, various approaches were proposed. For example, the winning system first performed source separation using unconditional universal sound separation (USS), and then applied a classification model to each separated signal to estimate its class [5]. The separation results were subsequently refined using the mixture, the separated signals, and the estimated class labels. In contrast, the second-place system conditioned the separation model on frame-level class estimates obtained by sound event detection (SED) [6]. Other proposed approaches included methods that directly estimated the classes of multiple sources from the mixture and then performed target sound extraction (TSE) based on the estimated classes [7], as well as approaches that jointly performed separation and classification with a single model [8].

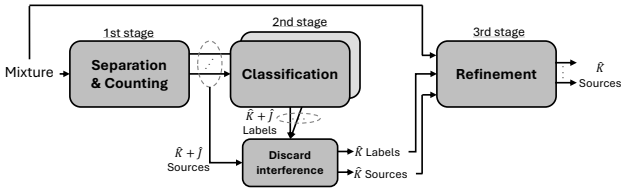

Figure 1: Overview of our system.

Our solution to this task is a three-stage separation and classification system, as illustrated in Fig. 1. First, the initial model takes a multichannel mixture as input and performs source separation and source-count estimation. Since the separation model is not trained to distinguish target from interfering sources, it estimates the total number of sources and separates a corresponding number of signals. Second, a classification model is applied to each separated signal to estimate its class. The classification model is designed to include an interference class, enabling it to determine whether an input signal corresponds to an interfering source. Third, for all sources classified as belonging to target classes in the second stage, we apply another separation model that takes the mixture, the separated signal, and the estimated class label as inputs, with the goal of improving the separation accuracy.

2. METHOD

We consider an M-channel signal $\boldsymbol{y} \in \mathbb{R}^{M\times L}$ as a mixture of K target sources $\boldsymbol{s}_k$, J interfering sources $\boldsymbol{v}_j$, and a noise $\boldsymbol{b}$:

$$\boldsymbol{y} = \sum_{k\in\mathcal{K}} \boldsymbol{s}_k + \sum_{j\in\mathcal{J}} \boldsymbol{v}_j + \boldsymbol{b}, \tag{1}$$

where $K = |\mathcal{K}| \in \{0,1,2,3\}$ and $J = |\mathcal{J}| \in \{0,1,2\}$. The sums over empty index sets are defined as zero. In DCASE 2026 Challenge Task 4, all signals are represented in first-order Ambisonics (FOA) format (i.e., $M = 4$). Target sources are defined as sound sources belonging to one of the 18 predefined classes, while interfering sources are sound sources from all other classes.

The goal of the S5 task is to separate K target sources on the W-channel from a mixture and estimate the class label for each source.

*Equal contribution.

Given that K varies across mixtures, a system is essentially required to perform the following three processes:

- Source separation: estimating the target sources
- Sound classification: estimating the class of each target source
- Source counting: estimating the number of target sources

Various approaches can be considered to achieve this goal. For example, the baseline system first directly estimates the class labels of all target sources from the mixture, and then performs separation using a model conditioned on the estimated class labels. Source counting is performed by the classification model by introducing a silence class.

In contrast, we considered it easier to estimate class labels after performing source separation. Therefore, we investigated a three-stage system as shown in Fig. 1: (i) source separation and counting, (ii) classification, and (iii) refinement. These models are described in the following sections.

2.1. First-stage model: Joint separation and counting

The first model jointly performs source separation and source-count estimation. Since the model is not trained to distinguish between target and interfering sources at this stage, it is trained to separate all non-background sounds, including interfering sources, and estimate their total number, $N := K + J$. The separation model is based on TF-Locoformer [9] with a spectral feature compression (SFC) [10] encoder and decoder. Although the original model is designed to separate a fixed number of sources, we extend it to enable source-count estimation and separation of a variable number of sources.

The SFC encoder takes the real and imaginary (RI) parts of a multichannel signal in the short-time Fourier transform (STFT) domain $\boldsymbol{Y} \in \mathbb{R}^{2M \times F \times T}$ as input, where F and T are the numbers of frequency bins and time frames, respectively, and the factor of 2 corresponds to the RI components. It then encodes them into a D-dimensional feature $\boldsymbol{Z} \in \mathbb{R}^{D \times F' \times T}$ while compressing the frequency sequence length to $F' (< F)$, using cross-attention with F' learnable queries $\boldsymbol{Q}^{\mathcal{E}} \in \mathbb{R}^{D \times F' \times 1}$ (time dimension is broadcasted):

$$\boldsymbol{Z} \leftarrow \texttt{SFCEncoder}(\boldsymbol{Y}, \boldsymbol{Q}^{\mathcal{E}}, \boldsymbol{A}^{\mathcal{E}}). \tag{2}$$

In cross-attention, a learnable positional bias $\boldsymbol{A}^{\mathcal{E}} \in \mathbb{R}^{H \times F' \times F}$, where H denotes the number of attention heads, is added to the attention scores before the softmax operation. The positional bias encourages the f'-th band of the encoded feature to preferentially attend to the frequency bins corresponding to the f'-th mel band.

Next, a learnable classification (CLS) token $\boldsymbol{C} \in \mathbb{R}^{D \times 1 \times T'}$, which will be used for source counting, is concatenated with the encoded features along the time-frame dimension, and the resulting sequence is processed by $B_1^{(1\mathrm{st})}$ TF-Locoformer blocks:

$$[\boldsymbol{C}, \boldsymbol{Z}] \leftarrow \texttt{TFLocoformer\_Blocks}([\boldsymbol{C}, \boldsymbol{Z}]) \tag{3}$$

The CLS token $\boldsymbol{C}$ is then fed into a source-count estimator based on multiple Conv1D layers. This estimator predicts the total number of target and interfering sources by solving a classification problem with six classes corresponding to counts from 0 to 5. Subsequently, a linear layer is applied to $\boldsymbol{Z}$ to estimate the number of features corresponding to the estimated source count $\hat{N}$:

$$\boldsymbol{Z} \leftarrow \texttt{Linear}_{\hat{N}}(\boldsymbol{Z}),\ \texttt{Linear}_{\hat{N}} : \mathbb{R}^{D \times F' \times T} \mapsto \mathbb{R}^{\hat{N}D \times F' \times T}. \tag{4}$$

Here, the model has five linear layers corresponding to $\hat{N} \in \{1, \ldots, 5\}$ with output size of $\hat{N}D$ [11]. By reshaping the output to $\boldsymbol{Z} \in \mathbb{R}^{\hat{N} \times D \times F' \times T}$, we obtain *internally disentangled* source features for the number of sources corresponding to the estimated count $\hat{N}$. Note that this linear layer and the subsequent blocks are not applied to samples for which $\hat{N} = 0$.

The $\hat{N}$ features are further processed by $B_2^{(1\mathrm{st})}$ TF-Locoformer blocks, where the weights of these TF-Locoformer blocks are shared across all sources, and the blocks are applied independently to each feature $\boldsymbol{Z}_n \in \mathbb{R}^{D \times F' \times T} (n \in \{1, \ldots, \hat{N}\})$:

$$\boldsymbol{Z}_n \leftarrow \texttt{TFLocoformer\_Blocks}(\boldsymbol{Z}_n). \tag{5}$$

Subsequently, the SFC decoder is applied to each feature $\boldsymbol{Z}_n$. The SFC decoder estimates spectrograms of separated signals $\hat{\boldsymbol{S}}_n \in \mathbb{R}^{2 \times F \times T}$ with sequence length F from features compressed to sequence length F' by cross-attention with F learnable queries $\boldsymbol{Q}^{\mathcal{D}} \in \mathbb{R}^{D \times F \times 1}$ and a positional bias $\boldsymbol{A}^{\mathcal{D}} \in \mathbb{R}^{H \times F \times F'}$:

$$\hat{\boldsymbol{S}}_n = \texttt{SFCDecoder}(\boldsymbol{Z}_n, \boldsymbol{Q}^{\mathcal{D}}, \boldsymbol{A}^{\mathcal{D}}). \tag{6}$$

2.2. Second-stage model: Classification on separated signals

Classification models are applied to each separated signal produced by the first-stage separation model to estimate its class label. Since separated signals include both target and interfering sources, we introduce an "Interference" class in addition to the 18 target classes, resulting in a 19-class classification problem. The cascade of the first-stage separation model and the second-stage classification model realizes all three required processes: target-source separation, class-label estimation, and source counting.

We build the classification models using five general-purpose audio representation models: BEATs [12], M2D [13], ASiT [14], PE-A-Frame-small [15], and PE-A-Frame-base [15]. The checkpoints for BEATs, M2D, and ASiT are provided through `PretrainedSED` [16][1],while those for PE-A-Frame are available on Hugging Face[2]. Each model consists of Transformer blocks (`Transformer`) preceded by a feature extractor (`Enc`), and we add a classification head (`Classifier`) as follows:

$$\hat{\boldsymbol{E}}_n = \texttt{Enc}(\hat{\boldsymbol{s}}_n), \tag{7}$$

$$\hat{\boldsymbol{V}}_n = \texttt{Transformer}(\hat{\boldsymbol{E}}_n), \tag{8}$$

$$\hat{\boldsymbol{o}}_n = \texttt{Classifier}(\hat{\boldsymbol{V}}_n), \tag{9}$$

where $\hat{\boldsymbol{s}}_n$ denotes the nth separated signal in the time domain. The classification head consists of two fully connected layers with global temporal average pooling between them. The feature extractor for M2D computes log-mel spectrograms, whereas BEATs and ASiT use similar filter-bank features. PE-A-Frame leverages a pretrained variational autoencoder called DAC-VAE [17].

The pretrained transformer blocks are fine-tuned together with the classification heads by minimizing the cross-entropy loss. For BEATs, M2D, and ASiT, we adopt mixup as data augmentation,

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

[2] `https://huggingface.co/facebook/pe-a-frame-base`

where the extracted features and target one-hot vectors are mixed with a random ratio $\alpha \sim \mathrm{Beta}(0.2, 0.2)$:

$$\hat{\boldsymbol{E}}_{\text{mix}} = \alpha \hat{\boldsymbol{E}}_{k_1} + (1-\alpha)\hat{\boldsymbol{E}}_{k_2}, \tag{10}$$

$$\boldsymbol{o}^{\star}_{\text{mix}} = \alpha \boldsymbol{o}^{\star}_{k_1} + (1-\alpha)\boldsymbol{o}^{\star}_{k_2}, \tag{11}$$

where $\boldsymbol{o}^{\star}_{k}$ denotes the true one-hot vector for the k-th source signal.

2.3. Third-stage model: Refinement

Although the S5 task can be addressed using the first- and second-stage models, we further attempt to improve the separation quality by applying an additional separation model, inspired by [5, 18]. The architecture of the refinement model is similar to that of the first-stage separation model, but it takes, in addition to the mixture, the separated signals estimated by the first-stage model and the class labels estimated by the second-stage model as auxiliary inputs.

Specifically, the input to the SFC encoder is augmented by concatenating the separated signals from the first-stage model to the multichannel mixture, and these signals are jointly encoded into D-dimensional features. Inspired by [18], instead of directly using the separated signals from the first-stage model, we use signals obtained by applying a multi-channel multi-frame Wiener filter (MCMFWF) to them. Instead of using a CLS token, the separator has a set of learnable prompts corresponding to the target source classes, each with a shape of $D \times 1 \times 1$. According to the class labels estimated in the second stage, the $\hat{K}$ corresponding learnable prompt vectors are concatenated with $\boldsymbol{Z}$, as in [19], with broadcasting along the frequency dimension. After processing by $B_1^{(\mathrm{3rd})}$ TF-Locoformer blocks, we obtain $\hat{K}$ disentangled features by taking the element-wise product between the transformed versions of the k-th learnable prompt $\boldsymbol{P}_k \in \mathbb{R}^{D \times F' \times 1}$ and the feature $\boldsymbol{Z} \in \mathbb{R}^{D \times F' \times T}$, with broadcasting along the time-frame dimension, instead of using a source-count-dependent linear layer as in Eq. (4). The remaining processing is the same as in the first-stage model: $B_2^{(\mathrm{3rd})}$ TF-Locoformer blocks and the SFC decoder are applied to each source.

We consider the following two variants of the refinement model. (i) MIMO system: all target sources and their class labels are provided as inputs, and all sources are refined simultaneously. (ii) SISO system: instead of processing all target sources simultaneously, each target source and its corresponding class label are provided together with the mixture, and the target sources are refined separately one by one. The MIMO system has the advantage of being able to perform refinement using information from all target sources. However, since the auxiliary information may contain errors in the separated signals and estimated class labels, the SISO system is expected to be less affected by such errors.

3. EXPERIMENTS

3.1. Experimental setup for separation models

For the first-stage model, we set $B_1^{(\mathrm{1st})} = 8$ and $B_2^{(\mathrm{1st})} = 4$, while for the third-stage model, we set $B_1^{(\mathrm{3rd})} = 6$ and $B_2^{(\mathrm{3rd})} = 3$. In both models, each TF-Locoformer block was configured with $D = 64$, $C = 384$, $K = 4$, $S = 1$, $H = 4$, $G = 8$, and an attention dimension of $E = 128$[3] in the first B_1 blocks. In the later B_2 blocks, C and E were changed to 256 and 96, respectively, while other parameters are unchanged. We use the no-positional-encoding (NoPE) variant of TF-Locoformer [20] in the first-stage model, while the original block with RoPE [21] is used in the third-stage model. The window and hop sizes of the STFT were set to 1024 and 512 samples, respectively, and F' was set to 48.

The separation models were trained in two stages: unsupervised pretraining followed by supervised fine-tuning. This strategy was motivated by previous studies showing that unsupervised pretraining is effective when only a limited amount of supervised data is available [22, 23]. We first trained the first-stage model with $B_1^{(\mathrm{1st})} = 6$ and $B_2^{(\mathrm{1st})} = 3$ on AudioSet [24] using MixIT [25] with the SNR loss. Since the number of sources contained in each mixture was unknown, the linear layer in Eq.(4) was configured to always output eight features. This pretraining was always performed using monaural inputs. During training, the batch size was set to 32, the input duration was set to 6 s, and the model was trained for 262.5k steps. AdamW [26] was used as the optimizer, with a fixed learning rate of 1e-3 and a weight decay of 1e-2.

Subsequently, the first-stage and third-stage models were initialized with the pretrained model and fine-tuned in a supervised manner using the DCASE 2026 Task 4 dataset [1]. Since some parameters, such as the CLS token in the first-stage model and the class-specific learnable prompts in the third-stage model, were not included in pretraining, they were randomly initialized. In addition, because the first-stage model has more blocks than the pretrained model, the first six B_1 blocks and the first three B_2 blocks were initialized with the pretrained weights. The seventh and eighth blocks in B_1 and the fourth block in B_2 were initialized by copying the weights of the sixth B_1 block and the third B_2 block of the pretrained model, respectively. During fine-tuning, the batch size was set to 32, the input duration was set to 10 s, and the models were trained for approximately 150k steps. The optimizer setup was the same as that used for pre-training, except that a cosine decay schedule was applied to the learning rate. During training, we used teacher forcing; the ground-truth number of sources and class labels were provided to the models.

In inference, for source-count estimation in the first stage, we considered not only using the prediction from a single model but also ensembling the outputs of multiple models. For the ensemble, we used a total of ten models trained with slightly different configurations, such as different lengths of the CLS token T', and averaged the output probabilities from their source-count estimators.

3.2. Experimental setup for classification model

Since the classification models are applied to separated signals, they should be robust to artifacts introduced by the first-stage separation process. To this end, we train the classification models not only on true target and interference source signals, but also on separated signals obtained from the first-stage separation models. Specifically, we use two separation models to separate the same mixtures and include the outputs from both models in the training data. Training with signals produced by different models exposes the classifiers to a wider variety of artifacts, which should improve their robustness.

To mitigate the data scarcity of the official dataset, we further augment the training data using external datasets. From AudioSet, we select clips that have only one label corresponding to either a target or an interference class. We use up to 1000 and 100 clips for each target and interference class, respectively. We also collect additional samples from the following datasets: "Alarm-

[3] Note that the notation here is consistent with that used in the original TF-Locoformer paper [9] and is independent of the notation defined in other sections of this paper.

Table 1: Evaluation results. Count. acc. denotes the source-counting accuracy of the first-stage model, CA-PI-SDRi denotes the CA-PI-SDRi for the final system output, and Acc. (mix) and Acc. (src) denote the mixture- and source-level classification accuracies, respectively.

	Validation				Dev-test		
System	Count. acc. [%]	CA-PI-SDRi [dB]	Acc. (mix) [%]	Acc. (src) [%]	CA-PI-SDRi [dB]	Acc (mix) [%]	Acc (src) [%]
Baseline [1]	-	8.26	56.56	67.99	8.49	60.71	70.39
System 1	**92.17**	14.25	**72.61**	**77.36**	14.84	**78.11**	**83.52**
System 2	**92.17**	14.36	**72.61**	**77.36**	14.77	**78.11**	**83.52**
System 3	**92.17**	**14.55**	**72.61**	**77.36**	**14.95**	**78.11**	**83.52**
System 4	91.83	14.37	71.50	75.67	14.41	74.00	80.76

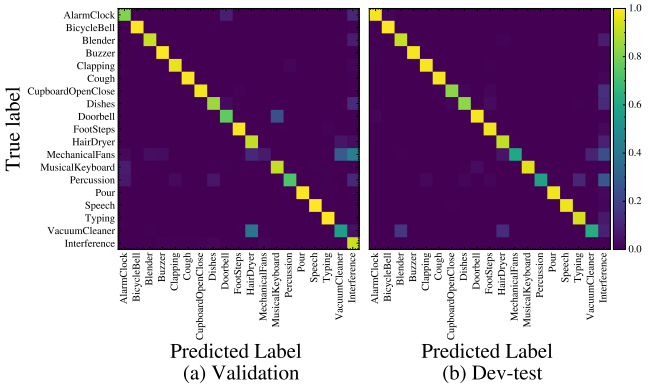

Figure 2: Confusion matrices for the BEATs-based classification model with row-wise normalization.

Clock" and "FootSteps" from NIGENS [27], "VacuumCleaner" from SINS [28], and "Clapping", "Footsteps", and "Pouring" from STARSS23 [29]. For SINS and STARSS23, we use the official metadata to extract the corresponding event segments. To ensure consistency with the pretraining stage, we resample the separated signals to 16 kHz for BEATs, M2D, and ASiT, and to 48 kHz for PE-A-Frame. In addition to mixup, we also apply frequency warping and filter augmentation to the features[4].

We fine-tune each model using AdamW [26] with 1000 warm-up steps. The peak learning rates are set to 4.0×10^{-4} for the classification head and 4.0×10^{-5} for the transformer blocks. To stabilize validation performance, we apply exponential moving average (EMA) to the model with a decay factor of 0.999, and the EMA checkpoint with the best F1 score mean over classes is used for inference. For the ensemble, we take the average across the five models' outputs after taking softmax.

3.3. Results

We evaluated the following four systems. All systems except System 4 used an ensemble of multiple model outputs for source counting and classification.

System 1: A system that uses only the first- and second-stage models, without applying the third-stage refinement.

System 2: The three-stage system described in Section 2, where the MIMO version is used for the third-stage refinement model.

System 3: A system in which the third stage of System 2 is replaced with the SISO version.

System 4: The same pipeline as System 3, but without ensembling the source-counting and classification results.

Table 1 shows the evaluation results on the validation and dev-test sets. Since the number of interfering sources in the dev-test set is not provided, counting accuracy is reported only for the validation set. Systems 1–3 use the same ensembled estimates for source counting and classification, and therefore have identical accuracies.

The results in Table 1 show that our system substantially improves performance over the baseline [1] in terms of both separation and classification. The first-stage model (System 1) already achieves reasonable counting accuracy and separation performance. When the refinement model is additionally applied, performance improvements are observed except for System 2 on the dev-test set, however, improvement is small. Possible reasons include the strong separation performance of the first-stage model, which benefits from unsupervised pretraining, and the larger mismatch between pretraining and fine-tuning in the refinement model, where the model structure differs more substantially, for example because the learnable prompts are randomly initialized and RoPE is additionally introduced. A detailed analysis of this behavior is left for future work. The results of Systems 3 and 4 show that ensembling multiple systems is effective for both counting and classification.

Figure 2 shows the class-wise confusion matrix for classification on separated signals[5] . The true label for each separated signal is assigned based on SNR with respect to the oracle target and interference signals. Since interference signals are not provided for the dev-test set, we ignore the separated signals that are not assigned to the oracle target signals, and thus the last row in Fig. 2 (b) is empty. The system achieves macro F1 scores of 85% on the validation set and 87% on the dev-test set. The classification errors are concentrated in a limited number of classes. This may be attributed to similarities in acoustic characteristics between certain classes (e.g., "HairDryer" and "VacuumCleaner").

4. CONCLUSION

This technical report described our solution to the DCASE 2026 Task 4 S5 task. The proposed system consists of three stages: joint separation and counting, classification, and refinement. We evaluated a total of four systems on the official DCASE 2026 Task 4 dataset, with differences in the use of ensembling and the design of the refinement model. The results showed that the proposed system with SISO refinement achieved the best performance. In future work, we plan to further improve the refinement model and conduct a more detailed ablation study.

[4]We follow data augmentation techniques implemented in `https://github.com/fschmid56/PretrainedSED`.

[5]Results on Fig. 2 are not based on separation results produced by our final USS model with ensemble for counting.

5. REFERENCES

[1] M. Yasuda, B. T. Nguyen, N. Harada, R. Serizel, M. Mishra, M. Delcroix, C. Hernandez-Olivan, S. Araki, D. Takeuchi, T. Nakatani, *et al.*, "Description and discussion on DCASE 2026 challenge task 4: Spatial semantic segmentation of sound scenes," *arXiv preprint arXiv:2604.00776*, 2026.

[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," in *Proc, EUSIPCO*. IEEE, 2025, pp. 266–270.

[3] M. Yasuda, B. T. Nguyen, N. Harada, R. Serizel, M. Mishra, M. Delcroix, S. Araki, D. Takeuchi, D. Niizumi, Y. Ohishi, T. Nakatani, T. Kawamura, and N. Ono, "Description and discussion on DCASE 2025 challenge task 4: Spatial semantic segmentation of sound scenes," 2025.

[4] B. T. Nguyen, M. Yasuda, D. Takeuchi, D. Niizumi, and N. Harada, "Class-aware permutation-invariant signal-to-distortion ratio for semantic segmentation of sound scene with same-class sources," *arXiv preprint arXiv:2601.22504*, 2026.

[5] Y. Kwon, D. Lee, D. Kim, and J.-W. Choi, "Self-guided target sound extraction and classification through universal sound separation model and multiple clues," DCASE2025 Challenge, Tech. Rep., June 2025.

[6] T. Morocutti, F. Schmid, J. Greif, P. Primus, and G. Widmer, "Transformer-aided audio source separation with temporal guidance and iterative refinement," DCASE2025 Challenge, Tech. Rep., June 2025.

[7] F. Wu and Z.-Q. Wang, "TS-TFGRIDNET: Extending TF-GRIDNET for label-queried target sound extraction via embedding concatentaiton," DCASE2025 Challenge, Tech. Rep., June 2025.

[8] Y. Nozaki, S. Sakurai, Y. Bando, K. Saijo, K. Imoto, and M. Onishi, "A hybrid S5 system based on neural blind source separation," DCASE2025 Challenge, Tech. Rep., June 2025.

[9] K. Saijo, G. Wichern, F. G. Germain, Z. Pan, and J. Le Roux, "TF-Locoformer: Transformer with local modeling by convolution for speech separation and enhancement," in *Proc. IWAENC*, 2024.

[10] K. Saijo and Y. Bando, "Input-adaptive spectral feature compression by sequence modeling for source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, 2026.

[11] J. Zhu, R. Yeh, and M. Hasegawa-Johnson, "Multi-decoder DPRNN: High accuracy source counting and separation," *arXiv preprint arXiv:2011.12022*, 2020.

[12] 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. ICML*, 2023, pp. 5178–5193.

[13] D. Niizumi, D. Takeuchi, M. Yasuda, B. Thien Nguyen, Y. Ohishi, and N. Harada, "M2D-CLAP: Exploring general-purpose audio-language representations beyond CLAP," *IEEE Access*, vol. 13, pp. 163 313–163 330, 2025.

[14] S. A. A. Ahmed, M. Awais, W. Wang, M. D. Plumbley, and J. Kittler, "ASiT: Local-global audio spectrogram vision transformer for event classification," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 3684–3693, 2024.

[15] A. Vyas, H.-J. Chang, C.-F. Yang, P.-Y. Huang, L. Gao, J. Richter, S. Chen, M. Le, P. Dollár, C. Feichtenhofer, *et al.*, "Pushing the frontier of audiovisual perception with large-scale multimodal correspondence learning," in *Proc. CVPR*, 2026, pp. 30 172–30 182.

[16] 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. ICASSP*, 2025.

[17] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, A. Vyas, B. Shi, C.-Y. Ma, C.-Y. Chuang, *et al.*, "Movie gen: A cast of media foundation models," *arXiv preprint arXiv:2410.13720*, 2024.

[18] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "TF-GridNet: Integrating full-and sub-band modeling for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 3221–3236, 2023.

[19] K. Saijo, J. Ebbers, F. G. Germain, G. Wichern, and J. Le Roux, "Task-aware unified source separation," in *Proc. ICASSP*, 2025.

[20] K. Saijo and T. Ogawa, "A comparative study on positional encoding for time-frequency domain dual-path transformer-based source separation models," *Proc, EUSIPCO*, 2025.

[21] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, "RoFormer: Enhanced transformer with rotary position embedding," *arXiv preprint arXiv:2104.09864*, 2021.

[22] A. Sivaraman, S. Wisdom, H. Erdogan, and J. R. Hershey, "Adapting speech separation to real-world meetings using mixture invariant training," in *Proc. ICASSP*, 2022, pp. 686–690.

[23] K. Saijo and Y. Bando, "Is MixIT really unsuitable for correlated sources? exploring MixIT for unsupervised pre-training in music source separation," in *Proc. WASPAA*, 2025, pp. 1–5.

[24] 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 *Proc. ICASSP*. IEEE, 2017, pp. 776–780.

[25] S. Wisdom, E. Tzinis, H. Erdogan, R. Weiss, K. Wilson, and J. Hershey, "Unsupervised sound separation using mixture invariant training," *Proc. NeurIPS*, vol. 33, pp. 3846–3857, 2020.

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2018.

[27] I. Trowitzsch, J. Taghia, Y. Kashef, and K. Obermayer, "The NIGENS general sound events database," *arXiv:1902.08314*, 2019.

[28] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, B. Van den Bergh, T. Van Waterschoot, B. Vanrumste, M. Verhelst, and P. Karsmakers, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," in *Proc. DCASE*, 2017.

[29] K. Shimada, A. Politis, P. Sudarsanam, D. A. Krause, K. Uchida, S. Adavanne, A. Hakala, Y. Koyama, N. Takahashi, S. Takahashi, T. Virtanen, and Y. Mitsufuji, "STARSS23: An audio-visual dataset of spatial recordings of real scenes with spatiotemporal annotations of sound events," in *Proc. NeurIPS*, vol. 36, 2023, pp. 72 931–72 957.