THE NU SYSTEMS FOR DCASE 2025 CHALLENGE TASK 2

Technical Report

Takuya Fujimura[1], *Ibuki Kuroyanagi*[1], *Tomoki Toda*[2]

[1] Graduate School of Informatics, Nagoya University, Nagoya, Japan
[2] Information Technology Center, Nagoya University, Nagoya, Japan

ABSTRACT

In this report, we present our anomalous sound detection (ASD) systems developed for DCASE 2025 Challenge Task 2. We propose a cascaded approach that integrates a target signal enhancement (TSE) model with a discriminative ASD system. First, we train the TSE model utilizing supplementary clean machine sounds and noise data. Then, we train the discriminative ASD system using the enhanced machine sounds to improve noise robustness. To further improve detection performance, we incorporate recently proposed techniques into the discriminative ASD system: multi-resolution spectrograms, pre-trained self-supervised learning features, and pseudo-label generation. Our final ensemble system has achieved 64.91% in the official scores calculated as a harmonic mean of the area under the curve (AUC) and partial AUC ($p = 0.1$) over all machine types and domains in the development set.

***Index Terms*—** anomalous sound detection, target signal enhancement, pseudo labels

1. INTRODUCTION

This report describes the systems we submitted for the DCASE 2025 Challenge Task 2 [1]. The task focuses on anomalous sound detection (ASD), which aims to detect mechanical failures from machine sounds. For system development, four requirements are imposed: (1) training models using only normal sounds, (2) addressing domain shifts, (3) training models for entirely new machine types, and (4) training models with or without attribute information. These requirements are identical to those of last year [2]. As a difference from last year, a supplementary dataset is newly provided, which includes clean machine sounds or noise data for each machine type. Participants can leverage this supplementary data to improve their system performance.

Our solution utilizes the supplementary data to construct a target signal enhancement (TSE) model (Fig. 1). The TSE model reduces noise in the machine sounds as a pre-processing, thereby improving the performance in the downstream ASD models. The downstream ASD models are based on the state-of-the-art discriminative methods [3], [4]. These methods train a feature extractor via classification of meta-information labels, and detect anomalies based on deviations from the normal training samples in the discriminative feature space. We employ both spectrum-based [3] and self-supervised learning (SSL) feature-based architectures [4] for the discriminative feature extractor, and ensemble their anomaly scores. Additionally, we adopt pseudo-label generation techniques [3] to effectively train the feature extractor for machine types lacking attribute information.

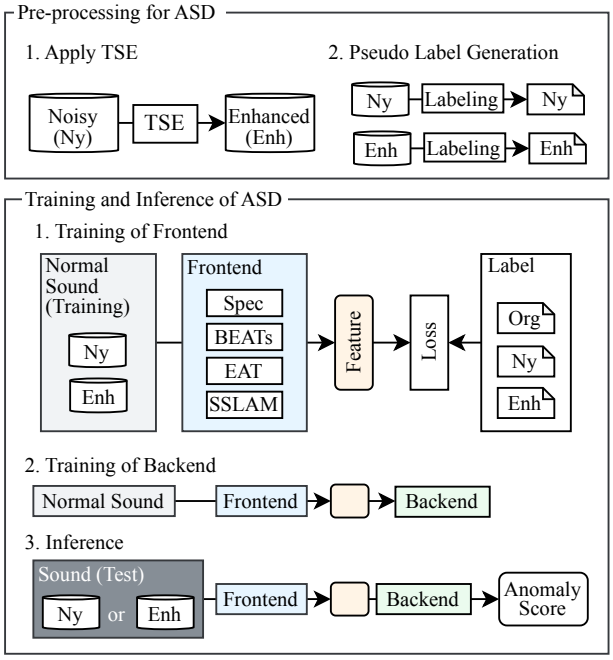

Figure 1: Overview of the proposed system

We conduct an experimental evaluation of our systems using the test data of the DCASE 2025 Challenge Task 2 development dataset [5], [6]. The results show that our systems significantly outperform the official baseline system and achieve the best performance when incorporating TSE pre-processing and pseudo-label generation techniques. Specifically, our system achieved 64.91% in the official scores, whereas the official baseline system [7] achieved 56.26%.

2. PROPOSED METHOD

Secs.2.1 and 2.2 describe the architectures of the proposed target signal enhancement (TSE) and anomalous sound detection (ASD) models, respectively. Sec.2.3 describes strategies for utilizing the TSE models in ASD tasks.

2.1. TSE Model

We construct a TSE model by utilizing the supplementary data. As shown in Fig. 2, we separately train the TSE model for each ma-

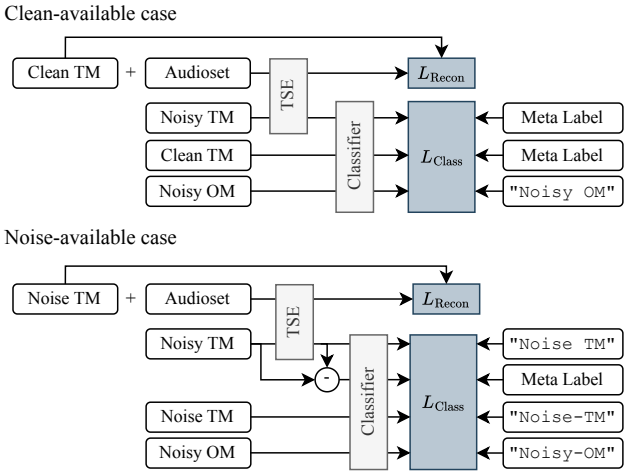

Figure 2: Training of the TSE model. The TSE model is trained separately for each machine type.

chine type using the following loss, L_{TSE}:

$$L_{\text{TSE}} = \lambda L_{\text{Recon}} + L_{\text{Class}}, \tag{1}$$

where λ is a hyperparameter that balances the two loss terms. The reconstruction loss L_{Recon} is defined as follows:

$$L_{\text{Recon}} = \mathcal{L}_D(\boldsymbol{x}_{\text{Target}}, f_{\text{TSE}}(\boldsymbol{x}_{\text{Target}} + \boldsymbol{n})), \tag{2}$$

where $\mathcal{L}_D(\cdot,\cdot)$ is an arbitrary reconstruction loss function, $f_{\text{TSE}}(\cdot)$ is the TSE model, $\boldsymbol{x}_{\text{Target}}$ is the target signal—either clean machine sounds or noise—provided in the supplementary data, and $\boldsymbol{n}$ is a sample drawn from AudioSet [8]. When $\boldsymbol{x}_{\text{Target}}$ is noise, the TSE model is trained to extract the noise component from the noisy machine sounds. Accordingly, the enhanced machine sounds are obtained by subtracting the estimated noise from the original noisy input.

L_{Class} is defined as $L_{\text{Class}}^{\text{Clean}}$ when clean machine sounds are available, and as $L_{\text{Class}}^{\text{Noise}}$ when noise signals are available:

$$\begin{aligned}
L_{\text{Class}}^{\text{Clean}} &= \mathcal{L}_C(f_{\text{Class}}(f_{\text{TSE}}(\boldsymbol{x}_{\text{NoisyTM}})), \boldsymbol{l}_{\text{Meta}}) \\
&\quad + \mathcal{L}_C(f_{\text{Class}}(\boldsymbol{x}_{\text{CleanTM}}), \boldsymbol{l}_{\text{Meta}}) \\
&\quad + \mathcal{L}_C(f_{\text{Class}}(\boldsymbol{x}_{\text{NoisyOM}}), \boldsymbol{l}_{\text{NoisyOM}}), \\
L_{\text{Class}}^{\text{Noise}} &= \mathcal{L}_C(f_{\text{Class}}(f_{\text{TSE}}(\boldsymbol{x}_{\text{NoisyTM}})), \boldsymbol{l}_{\text{NoiseTM}}) \\
&\quad + \mathcal{L}_C(f_{\text{Class}}(\boldsymbol{x}_{\text{NoisyTM}} - f_{\text{TSE}}(\boldsymbol{x}_{\text{NoisyTM}})), \boldsymbol{l}_{\text{Meta}}) \\
&\quad + \mathcal{L}_C(f_{\text{Class}}(\boldsymbol{x}_{\text{NoiseTM}})), \boldsymbol{l}_{\text{NoiseTM}}) \\
&\quad + \mathcal{L}_C(f_{\text{Class}}(\boldsymbol{x}_{\text{NoisyOM}})), \boldsymbol{l}_{\text{NoisyOM}}),
\end{aligned}$$

where $\mathcal{L}_C(\cdot,\cdot)$ is an arbitrary classification loss function, and $f_{\text{Class}}(\cdot)$ is a classifier. $\boldsymbol{x}_{\text{NoisyTM}}$, $\boldsymbol{x}_{\text{CleanTM}}$, and $\boldsymbol{x}_{\text{NoiseTM}}$ are the noisy machine sounds, clean machine sounds, and noise signals of the target machine type, respectively. $\boldsymbol{x}_{\text{NoisyOM}}$ is the noisy machine sounds of the other machine types. $\boldsymbol{l}_{\text{Meta}}$ is the meta-information label of machine type and attribute, while $\boldsymbol{l}_{\text{NoisyOM}}$ and $\boldsymbol{l}_{\text{NoiseTM}}$ are special labels assigned to each class. We use a frozen pre-trained BEATs model with a trainable linear classification head for $f_{\text{Class}}(\cdot)$ to encourage the TSE model to learn the denoising effect rather than relying on the classifier.

2.2. ASD Model

Our discriminative ASD model consists of a *frontend* and a *backend*. The frontend extracts features from the input machine sounds, while the backend computes anomaly scores based on the extracted features.

2.2.1. Frontend

We employ four different architectures for the frontend: Spec, BEATs, EAT, and SSLAM. Spec refers to an architecture that incorporates an amplitude spectrum and multi-resolution spectrograms as input features [3]. Spec independently transforms each input feature into a D_{Spec}-dimensional feature via neural networks. Subsequently, the D_{Spec}-dimensional features are concatenated to form a MD_{Spec}-dimensional feature, where M is the number of input features. Spec is trained from scratch using classification of meta-information labels. This architecture enables capturing anomalies from different perspectives, thereby improving ASD performance [3].

Based on the successful application of SSL models to the ASD task [4], [9], we also employ three SSL models: BEATs, EAT, and SSLAM. BEATs iteratively trains an acoustic tokenizer and an audio SSL model [10]. The SSL model is trained via a masked prediction task on discrete tokens generated by the tokenizer. The tokenizer is randomly initialized in the first iteration and then iteratively updated via knowledge distillation from the SSL model obtained in the previous iteration.

EAT is a SSL model based on the masked latent bootstrapping framework, in which a student model is trained via masked language modeling using the latent representations generated by a teacher model, and the teacher is continuously updated by the student [11]. To capture both global and local information, EAT combines utterance-level and frame-level reconstruction losses [11].

SSLAM refines the masked latent bootstrapping framework to enhance its ability to handle polyphonic sounds [12]. SSLAM trains a student model on mixtures so that it preserves the characteristics of the teacher model's representations for each individual source composing the mixture.

Following previous work [4], we fine-tune SSL models through a meta-information label classification task using low-rank adaptation (LoRA) [13]. From BEATs, we obtain a 768-dimensional feature sequence. We aggregate this feature sequence into a single representation using a statistics pooling layer [14], and project it to a D_{SSL}-dimensional feature using a linear layer. The resulting D_{SSL}-dimensional feature is used for both the classification task and the subsequent backend. For EAT and SSLAM, we obtain a 768-dimensional CLS feature, and project it to a D_{SSL}-dimensional feature using a linear layer.

Additionally, since the meta-label classification task may degrade performance for certain machine types [3], we also employ frozen pre-trained SSL models as frontends to improve robustness [15]. For BEATs, we average the output sequence to obtain a 768-dimensional feature, while for EAT and SSLAM, we directly use the 768-dimensional CLS feature.

2.2.2. Backend

We employ the same backend as in previous works [16]. The backend computes an anomaly score as the minimum cosine distance between an observation and the training data in the feature space.

To address the data imbalance between the source and target domains, SMOTE oversampling [17] is applied to the target-domain training data in advance.

2.2.3. Pseudo-label Generation

We employ pseudo-label generation techniques to effectively train the frontend for machine types without attribute information [3]. Unlike previous work [3], we utilize BEATs as the feature extractor for pseudo-label generation. First, we obtain a time-averaged 768-dimensional feature from the BEATs model, and then apply principal component analysis to reduce its dimensionality to 50. Finally, we perform Gaussian mixture model-based clustering to generate pseudo labels, where the number of clusters is determined by the Bayesian information criterion (BIC), with a maximum of eight clusters.

2.3. Strategies to utilize TSE for ASD

We utilize TSE during the training and inference stages of the ASD models, as well as for pseudo-label generation. At each stage, we leverage both noisy and enhanced machine sounds, taking into account the trade-off between noise robustness and the loss of machine sound components caused by TSE processing.

At the training and inference stages of the ASD models, we consider three strategies: (1) a baseline approach that uses the original noisy machine sounds for both training and inference; (2) an approach that uses enhanced machine sounds for both training and inference; (3) an approach that uses both noisy and enhanced machine sounds during training, but only noisy machine sounds during inference. In the third approach, we expect the ASD models to focus on machine sound components by jointly using enhanced machine sounds for the classification training.

We also utilize the enhanced machine sounds for pseudo-label generation. We separately use pseudo labels generated from noisy and enhanced machine sounds, as those generated from noisy sounds can reflect noise differences [3], while those from enhanced sounds may reflect the loss of machine sound components.

3. EXPERIMENTAL EVALUATIONS

3.1. Experimental setups

We conducted an experimental evaluation using the DCASE 2025 Challenge Task 2 development dataset (ToyADMOS2 [5], MIMII DG [6]) and the additional training dataset. The development dataset included training and test data of seven machine types: bearing, fan, gearbox, valve, slider, ToyCar, and ToyTrain. Also, the additional training dataset included training data for the other eight machine types. The training data included 1,000 samples of normal data for each machine type, with 990 samples from the source domain and 10 from the target domain. Furthermore, the supplementary data including 100 samples of either clean machine sounds or noise was provided for each machine type. The test data in the development dataset included 50 samples for each machine type, domain, and normal/anomalous class. Each recording was a 5 to 12-second single channel signal sampled at 16 kHz.

The architecture of the TSE model was a small-size TF-Locoformer [18] without positional encoding. For the short-time Fourier transform (STFT) used in the TF-Locoformer, the DFT size and frame shift were set to 512 and 128, respectively. λ was set to 0.5 and $\mathcal{L}_D$ was the negative signal-to-noise ratio (SNR) loss.

Table 1: Evaluation results. The values represent the harmonic mean of the official scores over all machine types. "Ny" and "Enh" indicate the noisy and enhanced machine sounds, respectively. In the "Label" column, "Ny" and "Enh" indicate pseudo labels generated from the noisy and enhanced machine sounds, respectively, while "Org" indicate the original labels. The last row shows the performance obtained by the frozen pre-trained SSL models.

Train	Test	Label	Spec	BEATs	EAT	SSLAM
Ny	Ny	Org	60.10	61.74	62.40	62.31
		Ny	61.77	64.14	63.69	63.34
		Enh	61.61	63.63	**63.95**	62.78
Enh	Enh	Ny	**62.63**	**64.54**	63.32	64.20
		Enh	62.36	64.37	63.87	**64.75**
Ny, Enh	Ny	Org	60.77	61.62	62.51	61.95
		Ny	61.86	63.84	63.82	63.55
		Enh	61.04	62.99	63.73	63.77
No	Ny	No		58.22	60.20	59.30

$\mathcal{L}_C$ was the the Sub-cluster AdaCos (SCAC) [19] with 16 trainable sub-cluster centers and a fixed scale parameter. We trained the TSE model for 2,400 epochs with a mini-batch size of 8 (i.e., 28,800 steps). Each sample was truncated or padded to 6 seconds. We used the AdamW optimizer [20] with gradient clipping at a maximum L_2-norm of 5. The learning rate was linearly increased from 0 to 0.0004 over the first 1,250 steps. The SNR for mixing $\boldsymbol{x}_{\mathrm{Target}}$ and $\boldsymbol{n}$ was randomly selected from the range $[-5, 5)$ dB. We used the TSE model except for fan, gearbox, BandSealer, and ToyRCCar.

For Spec of the ASD frontend, we used three multi-resolution spectrograms with DFT sizes of 256, 1024, and 4096. The frame shift was half of the DFT size, and frequency bins in the range of 200 Hz to 8000 Hz were used. The network consisted of the ResNet architecture similar to that in [21]. D_{Spec} was set to 128 and M was 4, resulting in a 512-dimensional feature. D_{Spec} was set to 128 and the number of input features M was 4, resulting in a 512-dimensional feature. We trained Spec for 16 epochs when using either the noisy or enhanced dataset, and for 8 epochs when jointly using both datasets. We used the AdamW optimizer with a fixed learning rate of 0.001 and a mini-batch size of 64. The loss function was the SCAC with 16 trainable sub-cluster centers and a fixed scale parameter. Mixup [22] was applied with a probability of 50%.

For SSL-based frontends, we used pre-trained checkpoints from their respective repositories: `BEATs_iter3.pt` for BEATs, `EAT-base_epoch10_pt.pt` for EAT, and `SSLAM_Pretrained/checkpoint_last.pt` for SSLAM. LoRA was applied to the query and key projection layers within the Transformer encoder for BEATs, and to the query, key, and value projection layers for EAT and SSLAM. For all SSL-based frontends, the LoRA rank was set to 64 and D_{SSL} was set to 256. We fine-tuned SSL models for 25 epochs with a mini-batch size of 8 (i.e., 46,875 steps). We used the AdamW optimizer, and the learning rate was linearly increased from 0 to 0.0001 over the first 5,000 steps. The loss function and mixup probability were the same as those used for Spec.

For SMOTE in the backend, we set the oversampling ratio to 20% and the number of neighbors to 2. For each system, we averaged anomaly scores across five different random seeds.

As a evaluation metric, we used the official scores, calculated as the harmonic mean of the area under the receiver operating charac-

Table 2: Evaluation results for the ensemble system combining Spec, BEATs, EAT, and SSLAM under each training and testing condition. The values represent the official scores. "hmean" indicates the harmonic mean of the scores over all machine types. "Ny" and "Enh" indicate the noisy and enhanced machine sounds, respectively. In the "Label" column, "Ny" and "Enh" indicate pseudo labels generated from the noisy and enhanced machine sounds, respectively, while "Org" indicate the original labels. The last row shows the performance obtained by the frozen pre-trained SSL models. * and † indicate machine types without attribute information and with supplementary clean machine sounds, respectively.

ID	Train	Test	Label	bearing*†	fan	gearbox	slider*	ToyCar†	ToyTrain*	valve†	hmean
①	Ny	Ny	Org	59.95	**54.54**	66.16	58.05	59.03	64.98	80.39	62.43
②			Ny	61.45	53.67	**69.15**	59.32	59.81	**67.22**	83.43	63.75
③			Enh	**70.44**	52.80	68.62	56.93	59.69	65.81	81.60	63.94
④	Enh	Enh	Ny	68.40	52.03	68.07	60.56	59.53	65.77	87.95	64.57
⑤			Enh	68.43	51.71	67.54	60.06	59.73	66.57	**89.11**	**64.58**
⑥	Ny, Enh	Ny	Org	57.05	53.38	66.63	**60.96**	59.24	64.85	82.45	62.44
⑦			Ny	65.14	53.75	67.48	60.89	58.93	65.77	84.36	64.09
⑧			Enh	67.08	51.78	65.72	59.62	59.18	65.39	82.97	63.38
⑨	No	Ny	No	55.51	51.78	55.68	58.98	**62.26**	66.90	79.54	60.44

Table 3: Evaluation results for the ensemble system combining different training and testing conditions. The values represent the official scores. "hmean" indicates the harmonic mean of the scores over all machine types. * and † indicate machine types without attribute information and with supplementary clean machine sounds, respectively.

Submission Name	ID	Ensemble	bearing*†	fan	gearbox	slider*	ToyCar†	ToyTrain*	valve†	hmean
Baseline (MSE)			59.75	49.90	55.26	55.68	56.73	55.73	62.42	56.26
Baseline (MAHALA)			61.45	51.34	58.61	57.58	55.87	48.37	56.37	55.34
Fujimura_NU_task2_1	⑩	(④+⑤)/2	69.31	51.92	68.19	**60.66**	59.54	66.37	88.94	64.85
	⑪	(⑥+⑦+⑧)/3	65.48	52.88	66.21	60.31	58.89	66.48	84.05	63.76
	⑫	(②+③+④+⑤)/4	**70.80**	52.54	**68.69**	59.45	59.79	66.75	86.22	**64.91**
Fujimura_NU_task2_2		0.75①+0.25⑨	59.77	**54.18**	65.40	58.23	59.18	65.79	80.92	62.44
		0.75⑩+0.25⑨	70.07	51.77	66.37	60.38	59.88	66.86	**89.08**	64.75
Fujimura_NU_task2_3		0.75⑪+0.25⑨	64.98	52.90	65.51	59.94	59.31	67.06	84.58	63.73
Fujimura_NU_task2_4		0.75⑫+0.25⑨	70.78	52.46	66.71	59.23	**59.99**	**67.40**	86.64	64.75

teristic (ROC) curve (AUC) and partial AUC (pAUC) with $p = 0.1$. The AUC was calculated for each domain using the normal samples from that domain and the anomalous samples from both domains, while the pAUC was calculated using samples from both domains.

3.2. Experimental results

Table 1 shows the harmonic mean of the official scores over all machine types for each frontend under each training and testing condition. First, we can see that Spec, BEATs, and SSLAM achieve their best performance when using enhanced machine sounds for both training and testing with pseudo labels. The effectiveness of pseudo labels is also observed under each training and testing condition. However, no significant improvement is observed by jointly using noisy and enhanced machine sounds for training in terms of the harmonic mean over machine types in the development dataset. Additionally, there is no consistent trend indicating whether pseudo labels generated from noisy or enhanced machine sounds lead to better performance. Finally, we also can see that SSL models outperform Spec in the development dataset.

Table 2 shows the official scores of the ensemble system combining Spec, BEATs, EAT, and SSLAM under the same training and testing conditions. The ensemble weights were set to $1/2$, $1/6$, $1/6$, and $1/6$ for Spec, BEATs, EAT, and SSLAM, respectively. We observe that systems ④ and ⑤ achieve high performance, significantly improving results for the bearing and valve machine types. Additionally, we can see that system ⑨ achieves competitive performance on ToyCar and ToyTrain without fine-tuning the frontend.

Table 3 shows the official scores of the ensemble system combining different training and testing conditions, compared with the official baseline systems [7]. Our final ensemble system significantly outperforms the official baseline.

4. CONCLUSION

In this report, we presented our systems for DCASE 2025 Challenge Task 2. First, we utilized the supplementary data to train a TSE model to improve downstream ASD performance. Second, we employed several state-of-the-art ASD frontends and combined them via ensemble of their anomaly scores. Third, we applied pseudo-label generation techniques to effectively train the frontends. The experimental results on the development dataset demonstrated the effectiveness of the proposed techniques, with our system achieving an official score of 64.91%.

5. ACKNOWLEDGMENT

This work was partly supported by JSPS KAKENHI Grant Number JP25KJ1439.

6. REFERENCES

[1] T. Nishida, N. Harada, D. Niizumi, *et al.*, "Description and discussion on DCASE 2025 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," *In arXiv e-prints: 2506.10097*, 2025.

[2] T. Nishida, N. Harada, D. Niizumi, *et al.*, "Description and discussion on dcase 2024 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2024, pp. 111–115.

[3] T. Fujimura, I. Kuroyanagi, and T. Toda, "Improvements of discriminative feature space training for anomalous sound detection in unlabeled conditions," in *Proc. ICASSP*, 2025, pp. 1–5.

[4] X. Zheng, A. Jiang, B. Han, *et al.*, "Improving anomalous sound detection via low-rank adaptation fine-tuning of pre-trained audio models," in *Proc. SLT*, 2024, pp. 969–974.

[5] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *Proc. DCASE*, 2021, pp. 1–5.

[6] K. Dohi, T. Nishida, H. Purohit, *et al.*, "Mimii dg: Sound dataset for malfunctioning industrial machine investigation and inspection for domain ggeneralization task," in *Proc. DCASE*, 2022.

[7] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "First-shot anomaly detection for machine condition monitoring: A domain generalization baseline," *Proceedings of 31st European Signal Processing Conference (EUSIPCO)*, pp. 191–195, 2023.

[8] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017, pp. 776–780.

[9] A. Jiang, B. Han, Z. Lv, *et al.*, "Anopatch: Towards better consistency in machine anomalous sound detection," in *Proc. Interspeech*, 2024, pp. 107–111.

[10] S. Chen, Y. Wu, C. Wang, *et al.*, "BEATs: Audio pre-training with acoustic tokenizers," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 5178–5193.

[11] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "Eat: Self-supervised pre-training with efficient audio transformer," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Main Track, 2024, pp. 3807–3815.

[12] T. Alex, S. Atito, A. Mustafa, M. Awais, and P. J. Jackson, "Sslam: Enhancing self-supervised models with audio mixtures for polyphonic soundscapes," in *International Conference on Learning Representations*, 2025.

[13] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models.," *ICLR*, vol. 1, no. 2, p. 3, 2022.

[14] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[15] P. Saengthong and T. Shinozaki, "Deep generic representations for domain-generalized anomalous sound detection," in *Proc. ICASSP*, 2025, pp. 1–5.

[16] A. Jiang, X. Zheng, B. Han, *et al.*, "Adaptive prototype learning for anomalous sound detection with partially known attributes," in *Proc. ICASSP*, 2025, pp. 1–5.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[18] 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 *2024 18th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2024, pp. 205–209.

[19] K. Wilkinghoff, "Sub-cluster adacos: Learning representations for anomalous sound detection," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

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

[21] K. Wilkinghoff, "Self-supervised learning for anomalous sound detection," in *Proc. ICASSP*, 2024, pp. 276–280.

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. ICLR*, 2018.