Discriminative Anomalous Sound Detection Using Pseudo Labels, Target Signal Enhancement, and Ensemble Feature Extractors

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*—We propose discriminative anomalous sound detection (ASD) systems designed to handle unlabeled data, noisy environments, and first-shot conditions. First, since discriminative methods suffer from significant performance degradation under unlabeled conditions, we generate pseudo labels to effectively train the discriminative feature extractors. Second, to improve noise robustness, we introduce a target signal enhancement (TSE) model as a pre-processing step. The TSE model is trained utilizing a small amount of clean machine sounds, together with a larger amount of noisy machine sounds. Third, to increase robustness across various machine types in first-shot conditions, we employ diverse architectures as feature extractors and ensemble their anomaly scores. Experimental results show that our systems achieve official scores of 64.85% and 59.99% on the DCASE 2025 development and evaluation sets, respectively, where the score is calculated as the harmonic mean of the AUC and partial AUC ($p = 0.1$) over all machine types and domains.**

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

1. INTRODUCTION

Anomalous sound detection (ASD) aims to identify abnormal machine behavior based on acoustic signals [1]–[5]. Due to the scarcity of anomalous sound data, ASD systems are typically trained using only normal machine sounds. ASD systems compute anomaly scores based on deviations from the normal sound distribution, assigning higher scores to sounds that are more likely to be anomalous.

State-of-the-art ASD methods are based on discriminative approaches [6]–[10]. This approach first trains a feature extractor to classify meta-information labels, such as machine type and operational status, associated with normal machine sounds. An anomaly score is then computed for each test sample by measuring its distance from normal training samples in the discriminative feature space. This discriminative space effectively captures differences in machine sounds, leading to high ASD performance.

While high ASD performance can be achieved with discriminative approaches, they still face several challenges in real-world applications, including the lack of meta-information labels, noisy environments, and first-shot conditions. First, although meta-information labels are essential for effectively training the feature extractor, they are sometimes unavailable in real-world scenarios, as collecting such labels requires human annotation or additional monitoring systems [5] Second, since ASD systems are typically deployed in factory environments, machine sounds are often contaminated by heavy noise, which can lead to misdetections. Third, it is desirable to develop ASD methods that can be applied to various machine types without relying on machine-specific knowledge including preliminary performance validation results for the target machine type (i.e., under first-shot conditions [4]).

In this paper, we propose a discriminative ASD system designed to address these challenges (Fig.1). First, to effectively train the feature extractor under unlabeled conditions, we adopt pseudo-label generation techniques [10]. Second, to mitigate performance degradation caused by noise, we introduce a target signal enhancement (TSE) model as a pre-processing step. The TSE model consists of a neural network trained via multi-task learning of reconstruction and classification losses using a small amount of clean machine sounds along with a larger amount of noisy machine sounds. Third, to improve generalization ability across various machine types, we employ diverse architectures as feature extractors and ensemble their anomaly scores. We conduct an experimental evaluation on the DCASE 2025 Challenge Task 2 dataset [11]. The results demonstrate that each component of our system is effective, and our system achieved fourth place in the official team rankings. Specifically, our system achieved official DCASE scores of 64.85% and 59.99% on the DCASE 2025 development and evaluation sets, respectively, whereas the official baseline system [12] achieved 56.26% and 56.51%.[1]

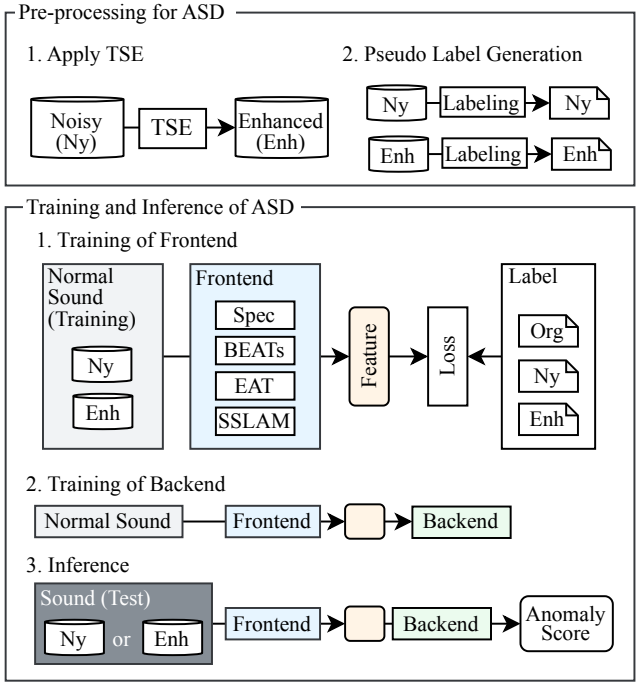

Fig. 1: Overview of the proposed system

2. PROPOSED METHOD

Sections 2.1 and 2.2 describe the architectures of the proposed TSE and ASD models, respectively. Section 2.3 presents strategies for utilizing the TSE model in the ASD task.

2.1. TSE Model

Following the DCASE 2025 Challenge Task 2 [11], we assume that a supplementary dataset is available for each machine type in

[1]The result on the evaluation set was obtained from the DCASE official website, https://dcase.community/challenge2025/task-first-shot-unsupervised-anomalous-sound-detection-for-machine-condition-monitoring-results

addition to the training dataset. The training dataset consists of noisy machine sounds, while the supplementary dataset contains a small amount of either clean machine sounds or machine-specific in-domain noise, where the noise characteristics differ across machine types. We construct TSE models by utilizing both the training and supplementary datasets.

The TSE model is separately trained for each machine type using the following multi-task loss L_{TSE}.

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

where λ is a hyperparameter that balances L_{Recon} and L_{Class}. L_{Recon} is defined as follows:

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

where $\mathcal{L}_D(\cdot,\cdot)$ is an arbitrary reconstruction loss function, $f_{\mathrm{TSE}}(\cdot)$ is the TSE model, $\boldsymbol{x}_{\mathrm{Target}}$ is the target signal–either a clean machine sound or in-domain noise–provided in the supplementary data, and $\boldsymbol{n}$ is out-domain noise drawn from AudioSet [13]. When $\boldsymbol{x}_{\mathrm{Target}}$ is in-domain noise, the TSE model is trained to extract the in-domain noise components. Accordingly, enhanced machine sounds are obtained by subtracting the estimated noise from the original noisy input.

L_{Class} is defined as $L_{\mathrm{Class}}^{\mathrm{Clean}}$ when clean machine sounds are available, and as $L_{\mathrm{Class}}^{\mathrm{Noise}}$ when noise signals are available in the supplementary data:

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

where $\mathcal{L}_C(\cdot,\cdot)$ is an arbitrary classification loss function, and $f_{\mathrm{Class}}(\cdot)$ is a classifier. $\boldsymbol{x}_{\mathrm{NoisyTM}}$ is the noisy machine sounds in the training dataset, while $\boldsymbol{x}_{\mathrm{CleanTM}}$ and $\boldsymbol{x}_{\mathrm{NoiseTM}}$ are the clean machine sounds and in-domain noise in the supplementary dataset, respectively. $\boldsymbol{x}_{\mathrm{NoisyTM}}$, $\boldsymbol{x}_{\mathrm{CleanTM}}$, and $\boldsymbol{x}_{\mathrm{NoiseTM}}$ all correspond to the target machine type, whereas $\boldsymbol{x}_{\mathrm{NoisyNM}}$ is the noisy machine sounds in the training datasets of non-target machine types. $\boldsymbol{l}_{\mathrm{Meta}}$ is the meta-information label, while $\boldsymbol{l}_{\mathrm{NoisyNM}}$ and $\boldsymbol{l}_{\mathrm{NoiseTM}}$ are single-class labels assigned to $\boldsymbol{x}_{\mathrm{NoisyNM}}$ and $\boldsymbol{x}_{\mathrm{NoiseTM}}$, respectively. This classification loss enables us to utilize the real noisy machine sounds $\boldsymbol{x}_{\mathrm{NoisyTM}}$ for training the TSE model, where $\boldsymbol{x}_{\mathrm{Target}} + \boldsymbol{n}$ in L_{Recon} is a synthetic mixture that includes out-of-domain noise $\boldsymbol{n}$. For the classifier $f_{\mathrm{Class}}(\cdot)$, we use a frozen pre-trained BEATs model [14] with a trainable linear classification head, encouraging the TSE model to focus on learning the denoising effect rather than relying on the classifier.

2.2. ASD Model

We describe the architecture of our discriminative ASD model. Hereafter, we refer to the discriminative feature extractor as the *frontend*, and the anomaly score computation module as the *backend*.

2.2.1. Frontend: To improve the robustness of the ASD model across various machine types, 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 [10]. Spec independently transforms each input feature into a D_{Spec}-dimensional feature via convolutional 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 multiple perspectives, thereby improving ASD performance [10].

Based on the successful application of self-supervised learning (SSL) models to the ASD task [8], [15], we employ three SSL models: BEATs [14], EAT [16], and SSLAM [17]. BEATs iteratively trains an acoustic tokenizer and a SSL model [14]. 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 latent representations generated by a teacher model, and the teacher is continuously updated by the student [16]. To capture both global and local information, EAT combines utterance- and frame-level reconstruction losses [16].

SSLAM refines the masked latent bootstrapping framework to enhance its capability in handling polyphonic sounds [17]. 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 [8], we fine-tune SSL models through a meta-information label classification task using low-rank adaptation (LoRA) [18]. From BEATs, we obtain a 768-dimensional feature sequence. We aggregate this feature sequence into a single representation using a statistics pooling layer [19], 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 anomaly score computation. 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 effectiveness of the meta-information labels depends on the machine type [10], we also employ frozen pre-trained SSL models as frontends to further improve robustness across various machine types [20]. For BEATs, we average the output sequence to obtain a 768-dimensional feature, whereas 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 [21]. The backend computes an anomaly score as the minimum cosine distance between an observation and the training data in the feature space. Since there is a data imbalance between the source and target domains, we apply SMOTE oversampling [22] to the limited training data in the target domain.

2.2.3. Pseudo-label Generation: We assume unlabeled conditions in the DCASE 2025 Challenge Task 2 [11], where the meta-information labels include machine type and attributes (e.g., machine operational status), but the attributes are unavailable for some machine types. To effectively train the frontend under the unlabeled conditions, we employ pseudo-label generation techniques [10]. The pseudo-label generation consists of two steps: extracting features from the training dataset and generating pseudo labels by applying clustering to the feature space. Although previous work used pre-trained PANNs [23] and OpenL3 [24] models as feature extractors for pseudo-label generation [10], we newly employ BEATs. Specifically, we average the 768-dimensional feature sequence extracted by the BEATs and then apply principal component analysis to reduce its dimensionality from 768 to 50. For the clustering, we use Gaussian mixture model, where

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	Org	60.29	64.43	62.62	63.16
		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

the number of clusters is determined by the Bayesian information criterion with a maximum of eight clusters.

2.3. Strategies to utilize TSE for ASD

We propose three strategies for using the TSE model on the ASD frontends during training and inference: (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 classification training, even though noisy machine sounds are used for inference.

We also utilize the TSE model for pseudo-label generation. We generate pseudo labels from noisy and enhanced machine sounds separately. Although pseudo labels that reflect noise differences can lead to performance degradation [10], we employ not only enhanced but also original noisy machine sounds for pseudo-label generation, taking into account the potential loss of machine sound components caused by the TSE processing.

3. EXPERIMENTAL EVALUATIONS

3.1. Experimental setups

We conducted an experimental evaluation using the DCASE 2025 Challenge Task 2 dataset, which partially includes ToyADMOS2 [25] and MIMII DG [26]. The dataset includes 15 machine types, each of which is assigned to either the development or evaluation subset. Specifically, the development set contains seven machine types (bearing, fan, gearbox, slider, ToyCar, ToyTrain, and valve), while the evaluation set contains eight (AutoTrash, HomeCamera, ToyPet, ToyRCCar, BandSealer, Polisher, ScrewFeeder, and CoffeeGrinder). For each machine type, training, supplementary, and test datasets are provided. The training dataset consists of 1,000 samples of normal machine sounds, with 990 samples from the source domain and 10 from the target domain. The supplementary dataset includes 100 samples of either clean machine sounds or in-domain noise. The test dataset contains 200 samples in total, including both normal and anomalous machine sounds from the source and target domains. Each recording is a 5 to 12-second single channel signal sampled at 16 kHz.

The architecture of the TSE model was a small-size TF-Locoformer [27] 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. For the loss function, λ was set to 0.5, and the reconstruction loss $\mathcal{L}_D$ was the negative signal-to-noise ratio (SNR) loss. The classification loss $\mathcal{L}_C$ was the Sub-cluster AdaCos (SCAC) [28] 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 [29] 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 manually inspected several samples of the enhanced machine sounds in the training dataset and decided to apply the TSE model except for fan, gearbox, BandSealer, and ToyRCCar because the TSE processing possibly degrades important machine sound components.

For Spec of the ASD frontend, we used three spectrograms with different DFT sizes of 256, 1024, and 4096, together with an amplitude spectrum. 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 [30] similar to that in [7]. The feature dimension D_{Spec} was set to 128 and the number of input features M was 4, resulting in a 512-dimensional feature vector. 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 [28] with 16 trainable sub-cluster centers and a fixed scale parameter. Mixup [31] 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 the feature dimension D_{SSL} was set to 256. We fine-tuned the 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 four different random seeds.

As an evaluation metric, we used the DCASE official scores, calculated as the harmonic mean of the area under the receiver operating characteristic (ROC) curve (AUC) and the 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 across all machine types for each frontend under each training and testing strategies using the TSE model. First, when original labels are used for training, BEATs and SSLAM significantly improve performance by using enhanced sounds for both training and testing, while Spec and EAT show almost no change in performance even if using the enhanced sounds. Additionally, training jointly using noisy and enhanced machine sounds slightly improves the performance of Spec when original labels are used. The effectiveness of pseudo labels is

Table 2: Evaluation results for each machine type. The values represent the official scores. In the "ID" column, Spec, BEATs, EAT, and SSLAM indicate individual systems, while ① to ⑩ indicate ensemble systems that combine Spec, BEATs, EAT, and SSLAM under the same training and testing strategy. "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. ⋆ indicate machine types without attribute information.

ID	Train	Test	Label	bearing⋆	fan	gearbox	slider⋆	ToyCar	ToyTrain⋆	valve	hmean
Spec	Ny	Ny	Org	57.93	51.12	62.10	55.85	59.55	62.34	78.10	60.10
BEATs				56.97	**59.73**	63.86	57.69	58.32	66.09	72.42	61.74
EAT				57.02	58.66	70.46	59.09	59.45	62.03	73.86	62.40
SSLAM				56.41	56.64	**71.97**	**62.41**	59.75	61.83	70.73	62.31
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	Org	60.04	51.97	65.35	62.06	58.31	66.81	83.18	62.81
⑤			Ny	68.40	52.03	68.07	60.56	59.53	65.77	87.95	64.57
⑥			Enh	68.43	51.71	67.54	60.06	59.73	66.57	**89.11**	**64.58**
⑦	Ny, Enh	Ny	Org	57.05	53.38	66.63	60.96	59.24	64.85	82.45	62.44
⑧			Ny	65.14	53.75	67.48	60.89	58.93	65.77	84.36	64.09
⑨			Enh	67.08	51.78	65.72	59.62	59.18	65.39	82.97	63.38
⑩	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 strategies. The values represent the official scores. "Dev" and "Eval" indicates the harmonic mean of the scores over machine types in the development and evaluation sets, respectively. The submission names are used for the DCASE 2025 Challenge Task 2 [11].

Submission Name	ID	Ensemble	Dev	Eval
`Fujimura_NU_task2_1`	⑪	(⑤+⑥)/2	64.85	59.99
	⑫	(⑦+⑧+⑨)/3	63.76	
	⑬	(②+③+⑤+⑥)/4	**64.91**	
`Fujimura_NU_task2_2`	⑭	0.75①+0.25⑩	62.44	58.51
`Fujimura_NU_task2_3`	⑮	0.75⑫+0.25⑩	63.73	59.34
`Fujimura_NU_task2_4`	⑯	0.75⑬+0.25⑩	64.75	59.91

consistently observed across all training and testing strategies. Spec, BEATs, and SSLAM achieve their best performance when we use both enhanced machine sounds and pseudo labels for training. There is no consistent trend regarding whether pseudo labels generated from noisy or enhanced machine sounds yield better results. Finally, SSL-based models consistently outperform Spec.

Table 2 shows the evalaution result for each machine type. The table includes the performance of the ensemble system that combines Spec, BEATs, EAT, and SSLAM under the same training and testing strategy. 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 ⑤ and ⑥, which use enhanced machine sounds and pseudo labels, achieve high performance, significantly improving results for the bearing and valve machine types. Additionally, system ⑩ shows competitive performance on ToyCar and ToyTrain without fine-tuning the SSL models. Finally, by comparing ensemble system ① with individual frontends, we find that ① consistently achieves high performance across all machine types, with harmonic mean scores that are higher than or similar to those of the individual frontends. This result highlights the effectiveness of ensembling multiple frontends under first-shot conditions.

Table 3 shows the official scores of the ensemble system combining different training and testing strategies. The following consistent performance ranking highlights the effectiveness of the TSE and pseudo labels: (1) ⑪ which uses enhanced machine sounds for both training and testing and also uses pseudo labels (ensemble of ⑤ and ⑥), (2) ⑯ which uses either noisy or enhanced machine sounds for both training and testing and also uses pseudo labels (ensemble of ②, ③, ⑤, ⑥, and ⑩), (3) ⑮ which jointly uses noisy and enhanced machine sounds for training and uses either original or pseudo labels (ensemble of ⑦, ⑧, ⑨, and ⑩) , and (4) ⑭ which uses neither TSE nor pseudo labels (ensemble of ① and ⑩).

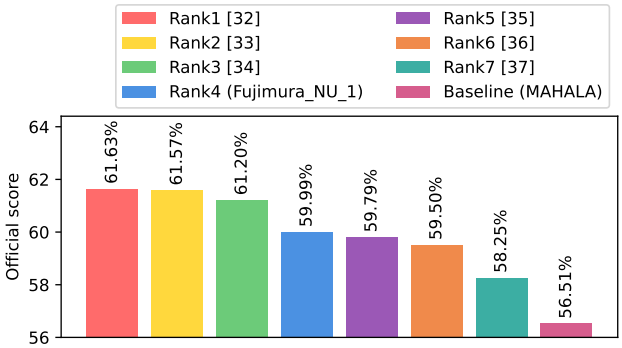

Fig. 2: Comparison of our system with the seven top-performing systems and the official baseline systems on the evaluation set of the DCASE 2025 Challenge Task 2.

Finally, Fig. 2 compares our system with the seven top-performing systems [32]–[37] and the official baseline systems [12] in the DCASE 2025 Challenge Task 2. Our system significantly outperforms the baseline systems and ranks fourth in the challenge.

4. CONCLUSION

In this paper, we proposed a discriminative ASD system to tackle the challenges of unlabeled data, noise, and first-shot conditions. First, we effectively utilized the unlabeled training data by generating pseudo labels through clustering in the BEATs feature space. Second, we introduced TSE models trained with a multi-task loss using both a small supplementary dataset and larger noisy training dataset. The TSE models were used as a pre-processing step for training, testing, and pseudo-label generation. Third, we ensembled multiple frontends—Spec, BEATs, EAT, and SSLAM—to handle diverse machine types. Experimental results showed that pseudo labels and TSE pre-processing significantly improved performance, and the ensemble of frontends outperformed individual frontends.

5. ACKNOWLEDGMENT

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

REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, *et al.*, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2020, pp. 81–85.

[2] Y. Kawaguchi, K. Imoto, Y. Koizumi, *et al.*, "Description and discussion on DCASE 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions," in *Proc. DCASE*, 2021, pp. 186–190.

[3] K. Dohi, K. Imoto, N. Harada, *et al.*, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," in *Proc. DCASE*, 2022, pp. 1–5.

[4] K. Dohi, K. Imoto, N. Harada, *et al.*, "Description and discussion on DCASE 2023 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2023, pp. 31–35.

[5] 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.

[6] K. Wilkinghoff, "Design choices for learning embeddings from auxiliary tasks for domain generalization in anomalous sound detection," in *Proc. ICASSP*, 2023, pp. 1–5.

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

[8] 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.

[9] I. Kuroyanagi, T. Hayashi, K. Takeda, and T. Toda, "Serial-oe: Anomalous sound detection based on serial method with outlier exposure capable of using small amounts of anomalous data for training," *APSIPA Transactions on Signal and Information Processing*, vol. 14, no. 1, pp. -, 2025.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM TASLP*, vol. 28, pp. 2880–2894, 2020.

[24] A. L. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Proc. ICASSP*, 2019, pp. 3852–3856.

[25] 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.

[26] 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.

[27] 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.

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

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

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

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

[32] L. Wang, "Pre-trained model enhanced anomalous sound detection system for dcase2025 task2," DCASE2025 Challenge, Tech. Rep., 2025.

[33] P. Saengthong and T. Shinozaki, "Genrep for first-shot unsupervised anomalous sound detection of dcase 2025 challenge," DCASE2025 Challenge, Tech. Rep., 2025.

[34] J. Yang, "A two stage fusion anomaly detection approach for task2," DCASE2025 Challenge, Tech. Rep., 2025.

[35] A. Jiang, W. Liang, S. Feng, *et al.*, "Thuee system for dcase 2025 anomalous sound detection challenge," DCASE2025 Challenge, Tech. Rep., 2025.

[36] X. Zheng, A. Jiang, B. Han, *et al.*, "Sjtu-aithu system for dcase 2025 anomalous sound detection challenge," DCASE2025 Challenge, Tech. Rep., 2025.

[37] S. Zhang, F. Xiao, S. Fan, Q. Zhu, W. Wang, and J. Guan, "Anomalous sound detection using pre-trained model with statistical feature difference representation," DCASE2025 Challenge, Tech. Rep., 2025.