MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Anomalous Sound Detection Meets Noise-Aware Self-Supervised Learning

Fujimura, Takuya; Wichern, Gordon; Masuyama, Yoshiki; Boeddeker, Christoph; Saijo, Kohei; Richter, Julius; Edo, Takahiro; Le Roux, Jonathan

TR2026-165 October 29, 2026

Abstract

In this paper, we introduce noise-aware self-supervised learning (NA-SSL) models for noise-aware anomalous sound detection (NA-ASD). NA-ASD is an ASD task with two-channel audio recordings, where one microphone is located close to the target machine and the other is located farther away to capture noise. For this task, we simulate two-channel recordings using diverse audio datasets and train NA-SSL models to extract clean SSL representations of the close-microphone signal by using the far-microphone recording dominated by background noise as auxiliary information. The NA-SSL models are then used as frontends in the standard ASD framework. Our experimental evaluation on the DCASE 2026 Challenge Task 2 development dataset demonstrates the effectiveness of the NA-SSL framework across three base SSL models (BEATs, EAT, and Dasheng), both with and without discriminative fine-tuning. Furthermore, the challenge results proved the effectiveness of the proposed approach, where the NA-BEATs system won the challenge by a large margin, achieving an official score of 70.24%, while the second-place system achieved 65.46%.

Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE) 2026

© 2026 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Anomalous Sound Detection Meets Noise-Aware Self-Supervised Learning

Takuya Fujimura[1,2*], *Gordon Wichern*[1], *Yoshiki Masuyama*[1], *Christoph Boeddeker*[1], *Kohei Saijo*[3], *Julius Richter*[1], *Takahiro Edo*[1], *Jonathan Le Roux*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, USA, [2]Nagoya University, Nagoya, Japan, [3]Information Technology R&D Center, Mitsubishi Electric Corporation, Kanagawa, Japan

***Abstract*—In this paper, we introduce noise-aware self-supervised learning (NA-SSL) models for noise-aware anomalous sound detection (NA-ASD). NA-ASD is an ASD task with two-channel audio recordings, where one microphone is located close to the target machine and the other is located farther away to capture noise. For this task, we simulate two-channel recordings using diverse audio datasets and train NA-SSL models to extract clean SSL representations of the close-microphone signal by using the far-microphone recording dominated by background noise as auxiliary information. The NA-SSL models are then used as frontends in the standard ASD framework. Our experimental evaluation on the DCASE 2026 Challenge Task 2 development dataset demonstrates the effectiveness of the NA-SSL framework across three base SSL models (BEATs, EAT, and Dasheng), both with and without discriminative fine-tuning. Furthermore, the challenge results proved the effectiveness of the proposed approach, where the NA-BEATs system won the challenge by a large margin, achieving an official score of 70.24%, while the second-place system achieved 65.46%.**

***Index Terms*—Anomalous sound detection, audio self-supervised learning, noise aware**

1. INTRODUCTION

Anomalous sound detection (ASD) aims to detect mechanical failures from machine sounds [1]–[8]. Since it is infeasible to exhaustively collect rare and diverse anomalous sounds, ASD systems are developed using only normal sounds. One major challenge in ASD is the presence of severe background noise in factory environments during both training data collection and testing. Motivated by this challenge, noise-aware ASD (NA-ASD) considers the use of auxiliary noise information [7]. NA-ASD employs a practical two-microphone recording setup, where one microphone is placed close to the target machine and the other is placed farther away. Although the far-microphone signal can also contain the target machine sound, the target sound is less dominant than in the close-microphone signal, making the far-microphone signal a useful auxiliary source of noise information. The NA-ASD task was newly introduced in DCASE 2026 Challenge Task 2 [7], and there is strong interest in developing ASD systems that can effectively utilize such auxiliary noise information.

A recent trend in standard ASD tasks is the use of audio self-supervised learning (SSL) models [8]–[13]. SSL models effectively capture subtle differences between normal and anomalous sounds, enabling training-free ASD by simply computing distances between a test sample and normal training samples in the representation space [9]–[11]. SSL models fine-tuned on machine sound datasets through classification tasks using machine-information labels capture more machine-specific information and achieve state-of-the-art performance [8], [12], [13]. However, under noisy conditions, such representations essentially encode irrelevant background noise as well, making it difficult to focus on target machine sounds. One possible solution is denoising training specialized for extracting specific target machine sounds, as in WavLM [14]. However, it is not applicable to ASD tasks because clean target machine sounds are unavailable as training data.

*This work was done during an internship at MERL.

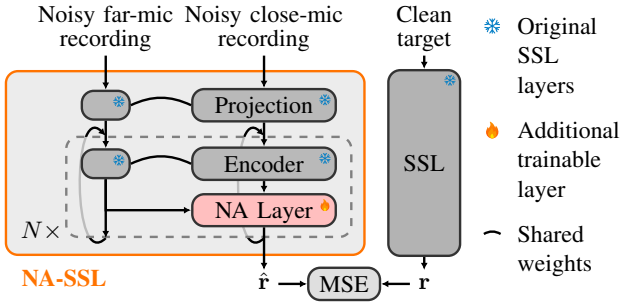

Fig. 1: Overview of NA-SSL, adapted from [15]. The model estimates clean SSL representations $\mathbf{r}$ from two-channel noisy signals, with one microphone placed close to the target sound source and the other placed farther away to capture background noise information.

A promising approach to handling noisy conditions in audio SSL models is the noise-aware SSL (NA-SSL) framework [15]. NA-SSL models are pre-trained on diverse audio signals to extract clean representations from noisy signals using auxiliary noise inputs. Because NA-SSL models perform conditional denoising based on the provided auxiliary noise information, they do not necessarily require clean target sounds or noise samples specific to the downstream task as pre-training data. That is, no clean machine sounds are used for NA-SSL training, only clean sounds from external datasets such as FSD50K [16]. In [15], one specific implementation of the NA-SSL framework was investigated, in which NA-BEATs was built on BEATs [17] and noise-only segments extracted from noisy input signals were used as auxiliary noise inputs. More generally, the framework is expected to accommodate any auxiliary signal that captures noise information, such as the far-microphone signal available in the NA-ASD task.

In this paper, we explore the use of NA-SSL models for the NA-ASD task. To this end, we simulate two-channel recordings using diverse audio signals and pre-train NA-SSL models to extract clean representations from two-channel noisy signals, as shown in Fig. 1. The NA-SSL models learn conditional denoising by exploiting the difference between the two channels: target sound information is more prominent in the close-microphone signal than in the far-microphone signal, whereas noise information is more prominent in the far-microphone signal. Through this conditional denoising scheme, NA-SSL models can provide denoised representations for the downstream ASD task without requiring clean target machine sounds as pre-training data. Furthermore, in addition to NA-BEATs, we newly design NA extensions of EAT [18] and Dasheng [19]. Experimental evaluations on the DCASE 2026 Challenge Task 2 development dataset show that NA-SSL substantially improves performance across the three base SSL models, both with and without discriminative fine-tuning. In the official challenge evaluation, NA-BEATs with discriminative fine-tuning achieved the best performance among 175 system submissions, with an official score of 70.24%, whereas the second-place system achieved 65.46% and the official baseline system achieved 59.80%.

2. SSL-BASED ASD

Typical ASD systems consist of a *frontend* and a *backend* [8]. The frontend extracts representations from machine sounds, and the backend computes anomaly scores based on distances in the representation space. Recent studies have extensively investigated audio SSL models as frontends [8]–[13], along with advances in backend techniques [9]–[11].

2.1. SSL models

BEATs [17] is an audio SSL model widely adopted for ASD [8]–[13]. It performs masked-prediction-based SSL in a discrete token space. A key feature of BEATs is its iterative training pipeline, in which the acoustic tokenizer and the SSL model are alternately refined through knowledge distillation. **EAT** [18] has also demonstrated strong ASD performance [8], [10], [11]. It is based on a masked bootstrapping framework, in which a student model is trained to reconstruct the latent representations of a teacher model at masked positions. It also introduces utterance-level losses and a CLS token to capture global information. **Dasheng** [19] is a recent SSL model based on a masked autoencoder framework and is trained on 272k hours of diverse audio data. Dasheng takes mel-spectrogram segments as input, whereas BEATs and EAT take mel-spectrogram patches as input. These models convert the input sequence into a representation sequence $\mathbf{r} \in \mathbb{R}^{L \times D}$, where L denotes the number of mel-spectrogram patches or segments and D denotes the representation dimension. The representation is then either directly used in the backend or further fine-tuned on machine sound datasets.

2.2. Discriminative fine-tuning

Fine-tuning is performed based on a classification task using machine-information labels [12], [13]. In recent DCASE datasets [5]–[7], the machine-information labels in the training data include the machine type (e.g., bearing), domain (source or target), and machine-dependent fine-grained attribute labels (e.g., rotation speed). Here, different attributes are included in the source and target domains. Also, some machine types do not have attribute labels. For such machine types, pseudo-labeling is employed [13], [20]. Pseudo attribute labels are generated by applying clustering to the representations of the training data. The pseudo attribute labels are then combined with the other available coarse labels of the target machine (i.e., the machine-type and domain labels) and the attribute labels of the other machines to form a multi-class classification task.

2.3. Backend techniques

In the backend, the SSL representation sequence $\mathbf{r}$ is aggregated to calculate the anomaly score. Saengthong et al. [9] proposed an aggregation technique that utilizes frequency information. This technique reshapes $\mathbf{r} \in \mathbb{R}^{L \times D}$ into $\mathbf{r}_{\mathrm{TF}} \in \mathbb{R}^{T \times F \times D}$, where T and F denote the numbers of mel-spectrogram patches along the time and frequency axes, respectively. It then applies average pooling along the time axis and concatenates the frequency axis to obtain a single FD-dimensional frequency-preserving representation. This representation was shown to improve backend performance compared with a global representation averaged over both the time and frequency axes [9].

BEAM [10] is another technique that builds on a frequency-preserving representation. It constructs a D-dimensional representation memory bank for each frequency index. An anomaly score is calculated separately for each frequency index and then averaged across the frequency indices to obtain a sample-level anomaly score. This technique mitigates undesirably high anomaly scores caused by variations within normal sounds. For example, an unseen combination of machine sound and noise can yield a high anomaly score, even when each frequency component has been separately observed in the training data. By avoiding strict joint comparison across frequencies, this technique can improve robustness to normal variations while retaining sensitivity to anomalies in each frequency.

Relative deviation pooling (RDP) [11] is a pooling technique designed as an alternative to simple average pooling. It computes a weighted average of the representation sequence, assigning each representation $\mathbf{r}_l$ a weight $w_l = (1+\hat{d}_l)^\gamma / \sum_{l'}(1+\hat{d}_{l'})^\gamma$ where $\hat{d}_l$ is the Euclidean distance between $\mathbf{r}_l$ and the sequence average, normalized by the maximum such distance within the sequence, and γ is a hyperparameter. This emphasizes representations that deviate from the average, thereby preserving temporally localized anomalies that are smoothed out by typical average pooling.

3. PROPOSED METHOD

We introduce NA-SSL models to the NA-ASD task. We follow the training procedure of the previous NA-BEATs implementation [15] but modify the auxiliary noise condition to match the NA-ASD task. Also, we design multiple NA-SSL models. Specifically, we design NA-BEATs, NA-EAT, and NA-Dasheng in the same manner based on BEATs (*BEATs_iter3.pt*), EAT (*EAT-base_epoch30_pretrain*), and Dasheng (*dasheng-base*), respectively.

3.1. Noise-aware self-supervised learning (NA-SSL)

Following [15], we train an NA-SSL model by distillation with a mean squared error (MSE) loss between the clean target representation sequence $\mathbf{r} \in \mathbb{R}^{L \times D}$ from the frozen original SSL model and the estimated representation sequence $\hat{\mathbf{r}} \in \mathbb{R}^{L \times D}$ [15], given by

$$\mathbf{r} = \mathrm{SSL}(\mathbf{s}), \tag{1}$$

$$\hat{\mathbf{r}} = \mathrm{NASSL}(\mathbf{x}, \mathbf{n}'), \tag{2}$$

where $\mathbf{s}$, $\mathbf{x}$, and $\mathbf{n}'$ denote clean target sound, noisy target sound, and auxiliary noise signal, respectively. For the NA-ASD task, we use the close-microphone signal as $\mathbf{x}$ and the far-microphone signal as $\mathbf{n}'$. We define the clean target sound $\mathbf{s}$ as the signal obtained at the close microphone using a room impulse response including early reflections within 50 ms. For NA-EAT, which produces a CLS token in addition to the representation sequence, we apply the MSE loss to both outputs with equal weights.

For all base SSL models, we construct NA extensions by inserting trainable NA layers into the frozen original SSL model after each Transformer layer, as in [15]. Here, BEATs, EAT, and Dasheng all have a projection layer and 12 Transformer layers. Both the close-microphone signal $\mathbf{x}$ and the far-microphone signal $\mathbf{n}'$ are passed through the original base SSL model. After each original Transformer layer, an NA layer refines the noisy target representation $\mathbf{z}_\mathbf{x} \in \mathbb{R}^{L \times D}$ using the corresponding noise representation $\mathbf{z}_{\mathbf{n}'} \in \mathbb{R}^{L \times D}$. Specifically, the cross-attention-based NA layer works as follows [15]:

$$\mathbf{z}_\mathbf{x} \leftarrow \mathbf{z}_\mathbf{x} + \mathrm{MHCA}(\mathrm{RMSNorm}(\mathbf{z}_\mathbf{x}), \mathrm{RMSNorm}(\mathbf{z}_{\mathbf{n}'})), \tag{3}$$

$$\mathbf{z}_\mathbf{x} \leftarrow \mathbf{z}_\mathbf{x} + \mathrm{FFN}_{\mathrm{SwiGLU}}(\mathrm{RMSNorm}(\mathbf{z}_\mathbf{x})), \tag{4}$$

where MHCA denotes multi-head cross-attention [21], with the first argument used as the query and the second argument used as the key and value. $\mathrm{RMSNorm}$ denotes root mean square normalization [22], and $\mathrm{FFN}_{\mathrm{SwiGLU}}$ denotes a Swish gated linear unit (SwiGLU)-based feed-forward network (FFN) [23]. The NA layer incorporates auxiliary noise information to refine the noisy representation and was shown to improve performance under noisy conditions in [15].

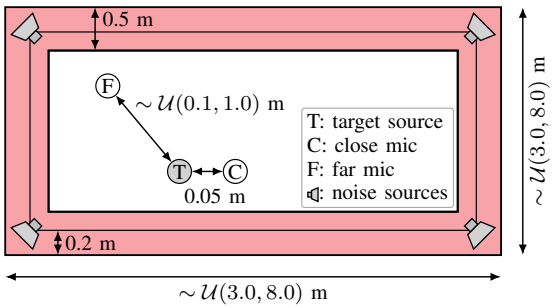

Fig. 2: Room geometry for simulation. The room height is sampled from $\mathcal{U}(1.5, 3.0)$ m, and all sources and microphones are placed at a height of 1.0 m.

3.2. Simulation of two-channel recordings

For training the NA-SSL models, we simulate two-channel recordings using diverse audio signals. FSD50K [16] is used as the target-sound dataset, while WHAM!48kHz [24], DEMAND [25], and QUT-NOISE [26] are used as the noise datasets. All signals are resampled to 16 kHz and randomly cropped or zero-padded to 10 s.

We simulate room impulse responses using Pyroomacoustics [27], as shown in Fig. 2. The reverberation time is randomly sampled from [0.1, 0.4] s. The target sound source is randomly placed with a 0.5 m margin from the walls. The close microphone is fixed 0.05 m from the target source, while the far microphone is randomly placed [0.1, 1.0] m away. Four different noise signals are randomly selected and played, each placed 0.2 m away from a corner of the room. The SNR at the close microphone is randomly selected from [-10, 10] dB.

3.3. ASD procedure using NA-SSL models

NA-SSL models can be combined with well-established SSL-based ASD techniques described in Section 2. For the ASD backend, we simply use the estimated clean representation $\hat{\mathbf{r}}$ instead of the noisy representation. Since the output of an NA-SSL model has the same format as that of the original SSL model, any backend technique designed for SSL representations can be directly applied to NA-SSL representations.

We can also perform discriminative fine-tuning of NA-SSL on top of $\hat{\mathbf{r}}$, where both the SSL model and the NA layers are updated. While the base SSL model is shared between the close- and far-microphone signals, the NA layers refine only the close-microphone representation. Therefore, fine-tuning the NA layers allows them to adapt specifically to the close-microphone representation, which predominantly contains the target machine sound.

4. EXPERIMENTAL EVALUATION

In the experimental evaluation, we demonstrate the general effectiveness of the NA-SSL framework across diverse settings, including three base SSL models, multiple backend techniques, and configurations with and without discriminative fine-tuning.

4.1. Setup

We conducted experimental evaluations using the DCASE 2026 Challenge Task 2 dataset, compiled from ToyADMOS2 [28] and MIMII DG [29]. The dataset consists of development and evaluation sets with distinct machine types. The development set includes seven machine types: bearingEmu, fan, gearboxEmu, valveEmu, sliderEmu, ToyCarEmu, and ToyCar. The evaluation set includes five machine types: ToyDrone, ToothBrush, SewingMachine, Sander, and BlowerDustCollector. Each machine type has 1000 training samples and 200 test samples. All training samples are normal, with 99% from the source domain and 1% from the target domain. Attribute labels are available only for the training data of the following machine types: fan, gearboxEmu, ToyCarEmu, ToyDrone, Sander, and BlowerDustCollector. The test data in the development set consist of 50 samples for each combination of domain and normal/anomalous class. The test data in the evaluation set are provided without domain and normal/anomalous labels for the official challenge evaluation. Each recording is a 6–16-second two-channel signal sampled at 16 kHz.

For the NA layers, we set the number of heads in MHCA to 8 and the hidden size of the FFN to 2304. We trained the NA-SSL models for 200 epochs using the AdamW optimizer, a fixed learning rate of 0.0001, and a batch size of 160. We applied exponential moving average (EMA) with a decay rate of 0.999 after 20k training steps.

For discriminative fine-tuning, we generated pseudo labels for machine types without attribute labels. When fine-tuning a given SSL or NA-SSL model, we generated the pseudo labels from that same model. The pseudo labels were generated by applying k-means clustering to the frequency-preserving representation. For each machine type, pseudo labels were generated in the source domain, where the number of clusters was set to 0.8% of the number of samples. For the target domain, all samples were assigned to a single target-domain class within each machine type.

During the discriminative fine-tuning, we aggregated the representation sequence $\mathbf{r}$ into a single 256-dimensional discriminative feature using an attentive statistics pooling layer [30] and a linear layer. In the backend, the SSL representation sequence before aggregation was used rather than the resulting discriminative feature. For the fine-tuning of the original SSL models, we applied low-rank adaptation (LoRA) with a rank of 64 to the query, key, and value projection layers in the attention modules. In NA-SSL, LoRA was also applied to MHCA in the NA layers with the same configuration. We fine-tuned the models for 25 epochs using sub-cluster AdaCos [31] with 16 sub-clusters, mixup with a probability of 0.5, AdamW optimizer, and a batch size of 8. The learning rate was linearly increased from 0 to 0.0001 over the first 5,000 steps.

We evaluated the performance with multiple backend techniques: frequency-preserving representation, BEAM, and RDP. To address the data imbalance between the source and target domains, we always used variance-minimization-based score rescaling [32] with four nearest neighbors. Cosine distance was used to calculate the anomaly score.

As the evaluation metric, we used the official score [7] calculated as the harmonic mean of three types of area under the receiver operating characteristic (ROC) curve (AUC): (a) partial AUC (pAUC) with $p = 0.1$, calculated using samples from both the source and target domains; and (b) source-domain and (c) target-domain AUCs, each calculated using normal samples from the corresponding domain and anomalous samples from both domains. We performed NA-SSL pre-training and discriminative fine-tuning for three trials with different random seeds. We report the mean and 95% confidence interval (CI95) of the official scores across the three trials and also report the ensemble result obtained by averaging the anomaly scores across the trials.

4.2. Results

Table 1 compares frontends using the same backend consisting of BEAM and RDP with $\gamma = 4$. First, we can see that NA-SSL is effective regardless of the base model and whether discriminative fine-tuning is applied. NA-SSL yields substantial improvements of over 20% for valveEmu and also improves performance for the other machine types. Discriminative fine-tuning further improves performance and achieves the best performance for all base models. Also, by using pseudo labels, the benefits of discriminative fine-tuning

Table 1: Evaluation of frontends across three base SSL models. The values are official scores, and Total shows the harmonic mean of the scores over the machine types. The mean, CI95, and ensemble results across the three trials are reported. The backend consists of BEAM and RDP with $\gamma = 4$. NA denotes NA-SSL pre-training, and Dis denotes discriminative fine-tuning. * denotes machine types without attribute labels.

		Mean (CI95)								Ensemble
Base	Method	bearingEmu*	fan	gearboxEmu	sliderEmu*	ToyCar*	ToyCarEmu	valveEmu*	Total	Total
BEATs	Original	63.27	52.73	64.63	64.42	**58.85**	58.70	69.68	61.32	61.32
	NA	62.33 (4.77)	52.19 (5.78)	63.73 (1.50)	69.59 (4.82)	58.45 (1.52)	**59.27** (3.27)	93.39 (3.97)	63.64 (1.71)	63.92
	Dis	63.38 (0.33)	51.47 (1.14)	66.47 (1.38)	66.43 (1.12)	55.30 (2.17)	57.15 (1.88)	80.55 (1.17)	61.78 (0.72)	61.82
	Dis NA	**64.64** (7.99)	**53.47** (8.31)	**69.23** (1.32)	**72.00** (2.60)	54.16 (4.13)	57.77 (2.70)	**95.63** (0.94)	**64.32** (0.77)	**64.57**
EAT	Original	**62.02**	53.16	61.48	61.18	54.99	**58.11**	65.96	59.27	59.27
	NA	61.21 (0.69)	56.17 (4.80)	64.06 (1.30)	66.63 (6.97)	**55.40** (1.26)	57.53 (0.17)	88.11 (1.97)	62.74 (1.41)	63.23
	Dis	61.63 (2.97)	48.30 (10.03)	62.61 (3.03)	58.24 (0.90)	50.45 (0.60)	57.71 (1.36)	66.54 (6.25)	57.21 (2.59)	56.72
	Dis NA	61.56 (2.18)	**57.60** (2.58)	**68.73** (2.16)	**71.10** (3.46)	54.27 (0.50)	55.87 (1.75)	**90.80** (3.70)	**63.88** (0.30)	**64.20**
Dasheng	Original	64.36	52.58	59.97	57.75	52.36	55.26	58.47	56.98	56.98
	NA	63.40 (2.14)	53.92 (10.09)	59.08 (8.00)	**65.65** (5.93)	**57.39** (1.74)	54.58 (2.74)	84.99 (13.63)	61.33 (2.13)	62.34
	Dis	65.25 (2.61)	52.95 (2.11)	**61.51** (1.02)	57.49 (1.83)	52.61 (0.55)	54.35 (2.67)	70.56 (3.39)	58.60 (0.69)	58.91
	Dis NA	**66.81** (3.51)	**54.81** (4.87)	60.82 (1.24)	64.87 (0.25)	52.93 (1.46)	**55.71** (3.03)	**90.03** (2.65)	**61.93** (2.13)	**63.10**

Table 2: Evaluation using multiple backends. The values are harmonic mean of the official scores over machine types in the development dataset. The values are reported as "Mean (CI95)". NA denotes NA-SSL pre-training, and Dis denotes discriminative fine-tuning. Global and Freq denote global and frequency-preserving representations, respectively. AP denotes the average pooling. The value in parentheses after "RDP" indicates γ. Since Dasheng extracts segment-wise representations, the frequency-preserving representation is identical to the global representation.

Base	Method	Global AP	Freq AP	Freq AP +BEAM	Freq RDP (4) +BEAM	Freq RDP (8) +BEAM
BEATs	Original	57.16	57.94	60.28	61.32	**62.02**
	NA	60.15 (1.08)	60.25 (2.74)	62.47 (1.88)	63.64 (1.71)	**64.10** (1.81)
	Dis	59.66 (0.66)	58.53 (0.72)	61.41 (0.52)	61.78 (0.72)	**61.91** (0.98)
	Dis NA	60.96 (2.42)	61.57 (0.98)	63.52 (1.39)	64.32 (0.77)	**64.35** (1.44)
EAT	Original	55.56	56.61	59.24	**59.27**	57.49
	NA	58.93 (0.75)	60.62 (1.01)	62.59 (1.50)	**62.74** (1.41)	62.42 (1.32)
	Dis	**59.21** (1.79)	56.99 (0.25)	58.69 (1.91)	57.21 (2.59)	55.88 (1.35)
	Dis NA	62.80 (1.46)	62.69 (0.63)	**63.99** (0.32)	63.88 (0.30)	63.12 (2.36)
Dasheng	Original	56.66	-	-	56.98	**57.07**
	NA	61.28 (3.71)	-	-	61.33 (2.13)	**61.75** (0.64)
	Dis	58.33 (0.42)	-	-	58.60 (0.69)	**59.07** (0.92)
	Dis NA	61.72 (2.53)	-	-	61.93 (2.13)	**62.63** (1.89)

can be observed even for machine types without attribute labels. We also observe high variance in ASD performance for NA-SSL models, highlighting the importance of ensembling multiple trials.

Table 2 shows the results obtained using different backend techniques. The results demonstrate the effectiveness of the individual backend techniques and show that their benefits also carry over to NA-SSL-based frontends. In particular, we can see that BEAM substantially improves performance. For RDP, larger values of γ yield greater improvements for BEATs and Dasheng, whereas RDP provides little benefit for EAT. These trends observed for RDP are consistent with the findings reported in [11].

We also conducted an ablation study to evaluate the effect of applying LoRA to the NA layers during discriminative fine-tuning, as shown in Table 3. Although discriminative fine-tuning remains effective even without LoRA in the NA layers, applying LoRA to the NA layers tends to yield better performance.

Figure 3 shows the results on the DCASE 2026 Challenge Task 2 evaluation dataset reported in [33]. It compares our Dis NA-BEATs

Table 3: Ablation study of discriminative fine-tuning. The values are harmonic mean of the official scores over machine types in the development dataset. The values are reported as "Mean (CI95)". The backend consists of BEAM and RDP with $\gamma = 4$. NA denotes NA-SSL pre-training, and Dis denotes discriminative fine-tuning.

Method	BEATs	EAT	Dasheng
NA	63.64 (1.71)	62.74 (1.41)	61.33 (2.13)
Dis NA	**64.32** (0.77)	**63.88** (0.30)	61.93 (2.13)
Dis NA w/o LoRA in NA layers	63.97 (1.81)	63.17 (1.17)	**62.09** (0.84)

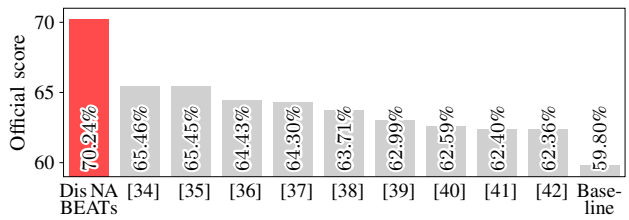

Fig. 3: Comparison of our Dis NA-BEATs with the ten top-performing systems and the official baseline system on the DCASE 2026 Challenge Task 2 evaluation dataset.

with the ten top-performing systems [34]–[42] and the official baseline system [7]. Here, the backend consists of BEAM and RDP with $\gamma = 4$, and the anomaly scores from three trials are ensembled. Dis NA-BEATs ranked first on the evaluation dataset with an official score of 70.24%, surpassing the second-place system (65.46%) and the official baseline system (59.80%) by a considerable margin. It is also noteworthy that our Dis NA-BEATs is built on only a single base SSL model with a single backend, whereas the other systems rely on ensembles of multiple SSL models [35], [40] and multiple backends [35], [38]–[40], [42]. Moreover, our Dis NA-BEATs outperforms the other BEATs-based systems [35], [39], [40], [42].

5. CONCLUSION

In this paper, we explored the use of NA-SSL models for the NA-ASD task. The NA-SSL models aim to extract clean SSL representations from noisy signals by leveraging auxiliary noise information captured by a microphone located farther from the target sound source. These models are first pre-trained on simulated two-channel recordings constructed from diverse audio signals. They are then used as frontends within a well-established SSL-based ASD framework. Our experimental results demonstrated the general effectiveness and broad compatibility of the NA-SSL framework across three base SSL models, multiple backend methods, and configurations both with and without discriminative fine-tuning.

REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda *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, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "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, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto *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, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "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, D. Albertini, R. Sannino, S. Pradolini, F. Augusti, K. Imoto, K. Dohi, H. Purohit *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] ——, "Description and discussion on DCASE 2025 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2025, pp. 55–59.

[7] T. Nishida, N. Harada, D. Takeuchi, D. Niizumi, K. Imoto, K. Dohi, H. Purohit, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2026 challenge task 2: Noise-aware unsupervised anomalous sound detection for machine condition monitoring," DCASE2026 Challenge, Tech. Rep., 2026.

[8] T. Fujimura, K. Wilkinghoff, K. Imoto, and T. Toda, "ASDKit: A toolkit for comprehensive evaluation of anomalous sound detection methods," in *Proc. DCASE*, 2025, pp. 40–44.

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

[10] ——, "Sub-band spectral matching with localized score aggregation for robust anomalous sound detection," *arXiv preprint arXiv:2603.13749*, 2026.

[11] K. Wilkinghoff, S. Yadav, and Z.-H. Tan, "Temporal pooling strategies for training-free anomalous sound detection with self-supervised audio embeddings," *arXiv preprint arXiv:2603.04605*, 2026.

[12] A. Jiang, B. Han, Z. Lv, Y. Deng, W.-Q. Zhang, X. Chen, Y. Qian, J. Liu, and P. Fan, "AnoPatch: Towards better consistency in machine anomalous sound detection," in *Proc. Interspeech*, 2024, pp. 107–111.

[13] B. Han, A. Jiang, X. Zheng, W.-Q. Zhang, J. Liu, P. Fan, and Y. Qian, "Exploring self-supervised audio models for generalized anomalous sound detection," *IEEE Trans. Audio, Speech, Lang. Process.*, 2025.

[14] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 6, pp. 1505–1518, 2022.

[15] T. Fujimura, Y. Masuyama, G. Wichern, C. Boeddeker, J. Richter, and J. Le Roux, "NABEATs: Noise-aware audio representation learning," in *Proc. IWAENC*, 2026, (accepted).

[16] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 829–852, 2022.

[17] 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*, vol. 202, 2023, pp. 5178–5193.

[18] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "EAT: Self-supervised pre-training with efficient audio transformer," in *Proc. IJCAI*, 2024, pp. 3807–3815.

[19] H. Dinkel, Z. Yan, Y. Wang, J. Zhang, Y. Wang, and B. Wang, "Scaling up masked audio encoder learning for general audio classification," in *Proc. Interspeech*, 2024.

[20] 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.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, vol. 30, 2017.

[22] B. Zhang and R. Sennrich, "Root mean square layer normalization," in *Proc. NeurIPS*, vol. 32, 2019.

[23] N. Shazeer, "GLU variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[24] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. Le Roux, "WHAM!: Extending speech separation to noisy environments," in *Proc. Interspeech*, 2019.

[25] J. Thiemann, N. Ito, and E. Vincent, "The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): A database of multichannel environmental noise recordings," in *POMA*, vol. 19, no. 1. Acoustical Society of America, 2013, p. 035081.

[26] D. Dean, S. Sridharan, R. Vogt, and M. Mason, "The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms," in *Proc. Interspeech*, 2010, pp. 3110–3113.

[27] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *Proc. ICASSP*, 2018, pp. 351–355.

[28] 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.

[29] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," in *Proc. DCASE*, 2022.

[30] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.

[31] K. Wilkinghoff, "Sub-cluster AdaCos: Learning representations for anomalous sound detection," in *Proc. IJCNN*, 2021, pp. 1–8.

[32] M. Matsumoto, T. Fujimura, W. Huang, and T. Toda, "Adjusting bias in anomaly scores via variance minimization for domain-generalized discriminative anomalous sound detection," in *Proc. DCASE*, 2025, pp. 25–29.

[33] DCASE Community, "DCASE 2026 challenge task 2: Noise-aware unsupervised anomalous sound detection for machine condition monitoring," https://dcase.community/challenge2026/task-first-shot-unsupervised-anomalous-sound-detection-for-machine-condition-monitoring, 2026, accessed: 2026-07-12.

[34] H. Wu, Z. Chang, P. Zhao, T. Zhao, Y. Zhang, M. Lei, and L. Zou, "Unsupervised anomalous sound detection via EAT and kalman-filtered near- and far-field dual-channel data," DCASE2026 Challenge, Tech. Rep., 2026.

[35] H. Yu, Y. Li, X. Fan, P. Li, Z. Lin, Z. Bai, and G. Huang, "Noise-aware unsupervised anomalous sound detection with pre-trained audio encoders and LoRA adaptation," DCASE2026 Challenge, Tech. Rep., 2026.

[36] F. Chu and M. Qian, "Anomalous sound detection system for DCASE 2026 task 2 using dual-channel spectral subtraction and efficient audio transformer," DCASE2026 Challenge, Tech. Rep., 2026.

[37] J. Yang, "Dual-channel attribute fine-tuning of pre-trained models for noise-aware anomalous sound detection," DCASE2026 Challenge, Tech. Rep., 2026.

[38] S. Zhang, W. Liang, X. Zheng, L. Xu, A. Jiang, T. Liu, P. Fan, W.-Q. Zhang, Y. Qian, X. Chen *et al.*, "THUEE system for DCASE 2026 task 2: Ensembling multiple anomaly detectors for generalized anomalous sound detection," DCASE2026 Challenge, Tech. Rep., 2026.

[39] P. Fan, A. Jiang, S. Zhang, L. Xu, W. Liang, T. Liu, X. Zheng, J. Li, W.-Q. Zhang, C. Lu *et al.*, "The WISTLAB system for DCASE 2026 task 2: Fine-grained scoring for noise-aware machine sound anomaly detection," DCASE2026 Challenge, Tech. Rep., 2026.

[40] C.-E. F. Krag, C. Rhod, E. Rosenlund, and S. B. Bræck, "Online and offline ensemble strategies for noise-aware unsupervised anomalous sound detection," DCASE2026 Challenge, Tech. Rep., 2026.

[41] H. Zhou, S. Chen, S. Chen, Z. Zheng, and S. Tang, "EAT token-MFS/GBC beam anomalous sound detection system for DCASE2026 task 2," DCASE2026 Challenge, Tech. Rep., 2026.

[42] A. Jiang, X. Zheng, W. Liang, S. Zhang, T. Liu, J. Liu, P. Fan, W.-Q. Zhang, C. Lu, X. Chen *et al.*, "AITHU submission for DCASE 2026 task 2: Robust machine-wise scoring with BEATs representations," DCASE2026 Challenge, Tech. Rep., 2026.