AISTAT LAB SYSTEM FOR DCASE 2026 TASK 2: NEAR-ANCHOR REPRESENTATION LEARNING WITH SELECTIVE FUSION

Technical Report

Min Jun Kim[1], *Jeong Ho Seo*[1], *Sang Chun Park*[1], *Seung Woo Sin*[2], *Changwon Lim*[1,2*]

[1]Department of Applied Statistics, Chung-Ang University, Seoul, Korea
[2]Department of Mathematics, Chung-Ang University, Seoul, Korea
{goodwill1669, jeongho0715, chunydp, hawar410, clim}@cau.ac.kr

ABSTRACT

This report describes the AISTAT LAB submission to DCASE 2026 Task 2 for noise-aware first-shot anomalous sound detection with synchronized near and far channel recordings under source and target domain shift. We propose a two-stage framework based on Efficient Audio Transformer encoders with channel-wise CMSN, attentive statistics pooling, and near-anchor DualPoolFusion. The fused representation is trained with Angular–Euclidean Compactness (AEC) using machine–domain–attribute labels. For inference, we combine KMeans-based cosine scoring, Relative Reference Distance Score (RRDS), and sample-wise adaptive RRDS weighting. The weighted ensemble achieves the best official score of 64.86% on the development-test set.

Index Terms— anomalous sound detection, first-shot learning, dual-channel audio, domain generalization, relative reference distance score

1. INTRODUCTION

Acoustic condition monitoring is an effective approach for detecting abnormal machine operation because microphones are inexpensive, non-intrusive, and easy to deploy in industrial environments [1]. However, real factory recordings often contain background noise, reverberation, and interference from nearby machines, making anomalous sound detection (ASD) difficult in practice [2]. Recent DCASE Task 2 settings have further emphasized practical deployment challenges, including source and target domain shift and first-shot adaptation with only a few target domain normal references [3, 4, 5]. DCASE 2026 Task 2 extends this problem to noise-aware ASD by providing synchronized near and far microphone recordings [6]. In this setting, the near channel generally captures target machine sounds more directly, whereas the far channel may contain complementary environmental information but can also introduce background-dominant noise. Therefore, an effective system should exploit dual-channel information selectively while preserving a machine-centered representation.

We propose a two-stage near-anchor dual-channel framework based on Efficient Audio Transformer encoders. Section 2 describes the representation learning, fusion, and scoring methods, and Section 3 reports the development-test results.

*Corresponding author.

Figure 1: Overview of the Stage-1 channel-specific representation learning pipeline.

2. PROPOSED METHOD

2.1. Overall Framework

Fig. 1 and Fig. 2 give an overview of the proposed two-stage dual-channel framework, which first learns channel-specific normal representations and then introduces controlled near–far interaction for anomaly scoring. In Stage 1, the near and far channels are processed independently by separate EAT encoders followed by attentive statistics pooling, producing 512-dimensional channel embeddings. In Stage 2, the near-channel encoder is kept fixed as an anchor, while the far-channel encoder and DualPoolFusion module are trained to selectively incorporate far-channel information through a sample-wise scalar gate and residual update.

2.2. Channel-wise cepstral mean and scale normalization

We apply channel-wise cepstral mean and scale normalization (CMSN) [7, 8] to the extracted log-mel features before encoder input. CMSN reduces channel and domain dependent scale mismatch in the input feature space. Unlike CMVN [9], which normalizes each feature dimension by its standard deviation, CMSN uses a robust percentile-based scale term to avoid excessive variance normalization while compensating for recording-condition mismatch.

Let x_c denote the log-mel feature of channel $c \in \{0, 1\}$, where $c = 0$ and $c = 1$ correspond to the near and far channels, respectively. The normalized feature is computed as

$$\tilde{x}_c = \frac{x_c - \mu_c}{s_c}. \tag{1}$$

For each channel c, domain $d \in \{\text{src}, \text{tgt}\}$, and mel bin f, we estimate the domain-wise mean and scale from normal

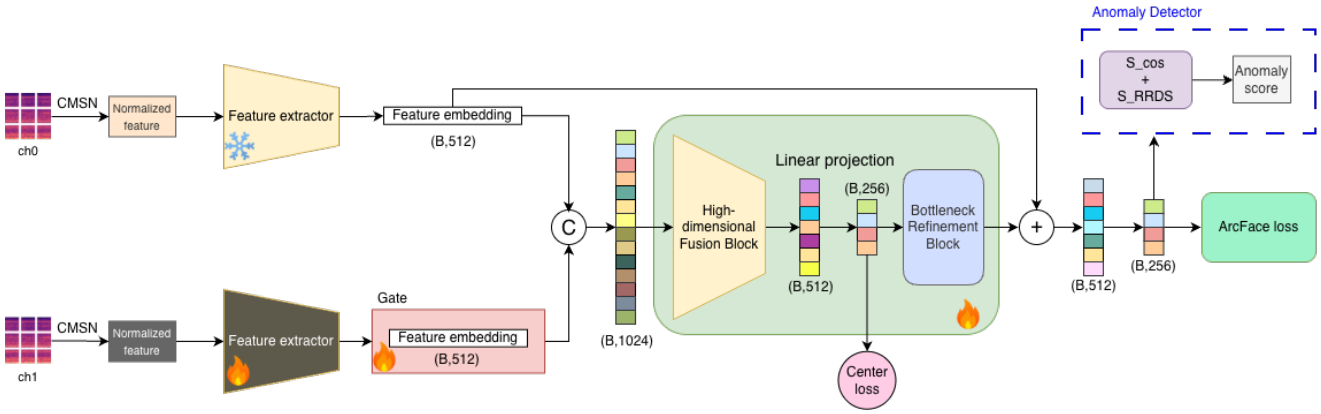

Figure 2: Overview of the Stage-2 near-anchor fusion and anomaly scoring pipeline.

reference clips:

$$\mu_c^d[f] = \mathrm{Mean}(x_c^d[:, f]),$$
$$s_c^d[f] = P_{95}(x_c^d[:, f]) - P_5(x_c^d[:, f]). \quad (2)$$

The final CMSN statistics are obtained by equally averaging the source and target domain estimates:

$$\mu_c[f] = \frac{1}{2}\mu_c^{\mathrm{src}}[f] + \frac{1}{2}\mu_c^{\mathrm{tgt}}[f],$$
$$s_c[f] = \frac{1}{2}s_c^{\mathrm{src}}[f] + \frac{1}{2}s_c^{\mathrm{tgt}}[f]. \quad (3)$$

The percentile range $P_{95} - P_5$ is used as a robust scale estimate, and source and target statistics are averaged equally to reduce source domain dominance. CMSN statistics are computed offline using only available normal training clips, including the provided first-shot target domain normal references. No test clips or anomaly labels are used.

2.3. Audio encoder backbone

We use the Efficient Audio Transformer (EAT) encoder [10] as the backbone feature extractor.

Each input is a 10-second dual-channel waveform sampled at 16 kHz. The near and far channels are processed separately rather than concatenated at the waveform level. For each channel, we remove the DC offset and extract 128-bin Kaldi-style log-mel filterbank features with a 10 ms frame shift. The extracted log-mel features are normalized by the CMSN procedure described above. Given a normalized log-mel sequence $\tilde{x}_c$ for channel $c \in \{0, 1\}$, the EAT encoder produces a sequence of 768-dimensional patch-level representations:

$$H_c = E_c(\tilde{x}_c), \quad H_c = \{h_{c,t}\}_{t=1}^{T}. \quad (4)$$

The sequence output is aggregated into a fixed-dimensional channel embedding using attentive statistics pooling in Section 2.4.

2.4. Channel-specific encoding and attentive statistics pooling

After CMSN normalization, each channel is encoded independently by its corresponding EAT encoder. For channel $c \in \{0, 1\}$, the encoder output is given by Eq. (4), where $h_{c,t} \in \mathbb{R}^{768}$ denotes the t-th patch-level representation. To convert this variable-length sequence into a fixed-dimensional clip-level embedding, we apply attentive statistics pooling (ASP) [11].

ASP computes attention weights $\alpha_{c,t}$ over temporal patches and summarizes the sequence using the weighted mean and weighted standard deviation:

$$\mu_c^{\mathrm{asp}} = \sum_{t=1}^{T} \alpha_{c,t} h_{c,t},$$
$$\sigma_c^{\mathrm{asp}} = \sqrt{\sum_{t=1}^{T} \alpha_{c,t}(h_{c,t} \odot h_{c,t}) - \mu_c^{\mathrm{asp}} \odot \mu_c^{\mathrm{asp}}}. \quad (5)$$

The two statistics are concatenated and projected to obtain a 512-dimensional channel embedding:

$$z_c = W_c[\mu_c^{\mathrm{asp}}; \sigma_c^{\mathrm{asp}}] + b_c, \quad z_c \in \mathbb{R}^{512}. \quad (6)$$

Here, z_0 and z_1 denote the near and far channel embeddings, respectively. These embeddings are learned independently in Stage 1 and are used as inputs to DualPoolFusion in Stage 2.

2.5. DualPoolFusion

DualPoolFusion combines z_0 and z_1 while keeping the near channel as the anchor. The far-channel contribution is controlled by a sample-wise scalar gate [12]:

$$g_1 = \sigma(\mathrm{MLP}([z_0; z_1; |z_0 - z_1|])), \quad z_1^{\mathrm{eff}} = g_1 z_1, \quad (7)$$

where $g_1 \in (0, 1)$ is broadcast to all dimensions of z_1. The gated far embedding is concatenated with the near embedding,

$$z_{\mathrm{cat}} = [z_0; z_1^{\mathrm{eff}}], \quad (8)$$

and passed through the fusion body:

$$h = F_A(\mathrm{LN}(z_{\mathrm{cat}})),$$
$$f_{256} = \mathrm{LN}(F_C(h)),$$
$$\delta = F_B(f_{256}),$$
$$z_{\mathrm{fusion}} = \mathrm{LN}(z_0 + \delta),$$
$$z = W_p z_{\mathrm{fusion}}. \quad (9)$$

Here, F_A is the high-dimensional fusion block, F_C maps the fused representation to a 256-dimensional bottleneck f_{256}, and F_B is the bottleneck refinement block that produces the residual update δ. The final embedding z is used for anomaly scoring in Section 2.7.

2.6. Loss function

We train the fused representation using an Angular–Euclidean Compactness (AEC) objective, which combines ArcFace loss and Center loss. Since the training data contain only normal samples, anomaly labels are not used during representation learning. Instead, each class label is defined by combining the machine type, domain, and attribute of a normal training clip. This label definition allows the model to learn compact normal clusters for each machine–domain–attribute condition while maintaining discriminative separation between different conditions.

The final scoring embedding z is optimized with ArcFace loss, which applies an additive angular margin to the target class logit [13]. For a mini-batch of size B, the ArcFace loss is defined as

$$\mathcal{L}_{\mathrm{arc}} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\big(s\cos(\theta_{y_i}+m)\big)}{\exp\big(s\cos(\theta_{y_i}+m)\big)+\sum_{j\neq y_i}\exp\big(s\cos\theta_j\big)}. \tag{10}$$

Here, y_i denotes the machine–domain–attribute label of the i-th normal sample, θ_{y_i} is the angle between the normalized embedding and the normalized target-class weight vector, m is the angular margin, and s is the scale factor. In our system, we set the ArcFace margin to $m = 0.5$ and the scale factor to $s = 64.0$.

To further reduce intra-class Euclidean variation, we apply Center loss [14] to the bottleneck representation f_{256}:

$$\mathcal{L}_{\mathrm{center}} = \frac{1}{2B}\sum_{i=1}^{B}\|f_{256,i}-c_{y_i}\|_2^2, \tag{11}$$

where c_{y_i} denotes the learnable center of class y_i. The final AEC loss is

$$\mathcal{L}_{\mathrm{AEC}} = \mathcal{L}_{\mathrm{arc}} + \gamma\mathcal{L}_{\mathrm{center}}. \tag{12}$$

We set the Center-loss weight to $\gamma = 0.01$. This objective jointly encourages angular separation in the final embedding space and Euclidean compactness in the bottleneck space.

2.7. Anomaly Scoring

After training, we extract fixed-dimensional embeddings from normal reference clips and test clips. Reference banks are constructed separately for each machine type, and each test embedding is compared only with source and target references from the same machine type. Source domain normal embeddings are summarized by KMeans centers [15], while target domain normal embeddings are kept as raw first-shot references:

$$C_m = \mathrm{KMeans}(Z_m^{\mathrm{src}}, K), \quad T_m = Z_m^{\mathrm{tgt}}, \tag{13}$$

where C_m and T_m denote the source and target reference banks for machine type m, respectively.

The cosine distance between two embeddings z and r is

$$d_{\cos}(z,r) = 1 - \frac{z^\top r}{\|z\|_2\|r\|_2}. \tag{14}$$

The primary anomaly score is the minimum cosine distance to either source centers or target references:

$$S_{\cos}(z) = \min\left(\min_{c\in C_m} d_{\cos}(z,c), \min_{t\in T_m} d_{\cos}(z,t)\right). \tag{15}$$

We define a Relative Reference Distance Score (RRDS), which measures the nearest-reference Euclidean distance relative to a reference-bank scale. For the source side, $d_1^{\mathrm{src}}(z)$ is the distance from z to its nearest source center, and $d_2^{\mathrm{src}}(z)$ is the distance from that center to its nearest neighboring center:

$$S_{\mathrm{RRDS}}^{\mathrm{src}}(z) = \frac{d_1^{\mathrm{src}}(z)}{d_1^{\mathrm{src}}(z)+d_2^{\mathrm{src}}(z)}. \tag{16}$$

For the target side, we use the median in-bank 1-nearest-neighbor distance α^{tgt} as a robust scale:

$$S_{\mathrm{RRDS}}^{\mathrm{tgt}}(z) = \frac{d_1^{\mathrm{tgt}}(z)}{d_1^{\mathrm{tgt}}(z)+\alpha^{\mathrm{tgt}}}. \tag{17}$$

The auxiliary RRDS score is

$$S_{\mathrm{RRDS}}(z) = \min\left(S_{\mathrm{RRDS}}^{\mathrm{src}}(z), S_{\mathrm{RRDS}}^{\mathrm{tgt}}(z)\right). \tag{18}$$

Fixed RRDS fusion is defined as

$$S_{\mathrm{fixed}}(z) = S_{\cos}(z) + \lambda S_{\mathrm{RRDS}}(z), \tag{19}$$

where $\lambda = 1.0$. For sample-wise adaptive RRDS weighting, for domain $d \in \{\mathrm{src}, \mathrm{tgt}\}$, let $d_1^d(z)$ be the distance from z to its nearest raw reference and $d_2^d(z)$ be the nearest-neighbor distance of that reference within the same raw reference bank:

$$h^d(z) = \frac{d_1^d(z)}{d_1^d(z)+d_2^d(z)}, \quad W(z) = \frac{h^{\mathrm{src}}(z)}{h^{\mathrm{src}}(z)+h^{\mathrm{tgt}}(z)}. \tag{20}$$

The adaptive score is

$$S_{\mathrm{final}}(z) = S_{\cos}(z) + W(z)S_{\mathrm{RRDS}}(z). \tag{21}$$

3. EXPERIMENTAL SETUPS AND RESULTS

3.1. Experimental setup

We evaluate all submitted systems on the DCASE 2026 Task 2 development-test set [6], which contains seven machine types with synchronized near and far channel recordings. For each machine type, source and target domain test samples are evaluated separately.

All models are trained on normal clips from the development and additional training datasets [16, 17]. To compare the effect of including target domain normal references during training, we train models under two data configurations: source-only and source-plus-target.

Following the official evaluation protocol [6], we report source domain AUC, target domain AUC, pAUC, and the official score, defined as the macro-averaged harmonic mean of these three metrics over machine types.

The models are optimized with Adam using a cosine warm restarts learning-rate scheduler [18] and gradient accumulation. Unless otherwise stated, the EAT encoders were initialized from the publicly available EAT pre-trained checkpoint.

Machine	System 1	System 2	System 3	System 4
ToyCar				
AUC source	**75.72**	58.20	69.36	72.24
AUC target	**89.48**	87.76	89.40	88.88
pAUC	**63.74**	60.84	62.37	62.68
ToyCarEmu				
AUC source	62.68	**69.24**	62.64	64.40
AUC target	84.64	**86.76**	83.96	85.88
pAUC	59.37	**60.11**	56.89	58.32
bearingEmu				
AUC source	**62.40**	60.24	55.96	57.48
AUC target	58.40	58.68	59.16	**59.24**
pAUC	56.63	54.47	**59.95**	58.63
fan				
AUC source	**76.64**	68.76	72.36	75.36
AUC target	53.84	55.24	**55.40**	54.16
pAUC	55.26	**57.00**	54.63	54.84
gearboxEmu				
AUC source	71.48	64.24	**75.20**	74.28
AUC target	63.88	**68.00**	64.60	65.12
pAUC	53.05	53.16	**55.53**	54.89
sliderEmu				
AUC source	69.08	64.04	**69.36**	69.16
AUC target	58.60	**61.60**	57.00	58.00
pAUC	**53.95**	51.16	53.84	52.74
valveEmu				
AUC source	76.00	83.16	81.24	**86.04**
AUC target	76.00	75.96	75.16	**77.20**
pAUC	60.58	62.53	63.89	**67.47**
All (Avg)				
AUC source	70.10	66.05	68.56	**70.29**
AUC target	66.94	**68.51**	67.06	67.44
pAUC	57.29	56.76	57.93	**58.14**
Official	64.30	63.35	64.16	**64.86**
Scoring	$S_{\cos}$	$S_{\cos}$ $+ S_{\mathrm{RRDS}}$	$S_{\cos}$ $+ W(z)S_{\mathrm{RRDS}}$	Ensemble

Table 1: Development-test set results by machine and submitted system scoring.

3.2. Submitted Systems and Results

We submitted four systems based on the same framework (Section 2) with different scoring or ensemble configurations. System 1 denotes the $S_{\cos}$ model trained on source domain normal data only, whereas System 1(S+T) denotes the corresponding cosine-scoring model trained on both source domain normal clips and first-shot target domain normal references. System 2 uses fixed RRDS fusion and System 3 uses adaptive RRDS weighting. System 1 was submitted as an individual system, but System 4 did not include it as an ensemble component. Instead, System 4 combines System 1(S+T), System 2, and System 3; we used System 1(S+T) because it achieved the best source domain AUC among the candidates evaluated under different training data configurations (Table 3).

Table 1 summarizes the development-test results. Among the single-model systems, System 1 achieves the highest official score, System 2 improves target domain AUC at the cost of source domain AUC, and System 3 achieves the best pAUC. System 4 leverages these complementary strengths and achieves the best overall official score of 64.86.

The ensemble weights were determined by grid search on the development-test set. For each candidate weight combination, the final anomaly score was computed as a weighted sum of the component anomaly scores, and the combination that maximized the official final score was selected. The ensemble weights are shown in Table 2.

Model	System 1(S+T)	System 2	System 3
Weight	0.286	0.129	0.585

Table 2: Ensemble weights of the component models used in System 4.

Train data	Model	AUC source	AUC target	pAUC	Official score
Source + Target	near-ch encoder(S+T)	68.32	66.29	56.08	63.09
	far-ch encoder(S+T)	58.72	54.60	51.87	54.92
	System 1(S+T)	**71.13**	64.61	55.71	63.17
	System 2(S+T)	67.90	66.77	56.65	63.35
	System 3(S+T)	68.81	66.60	56.32	63.42
Source	near-ch encoder	69.91	65.69	55.12	62.93
	far-ch encoder	58.80	54.03	51.83	54.73
	System 1	70.10	66.94	57.29	**64.30**
	System 2	66.05	**68.51**	56.76	63.35
	System 3	68.56	67.06	**57.93**	64.16

Table 3: Development-test set results of different models according to training data.

4. CONCLUSION

We presented the AISTAT LAB system for DCASE 2026 Task 2 on noise-aware first-shot anomalous sound detection with synchronized near and far microphone recordings. The proposed two-stage framework first learns channel-specific representations using separate EAT encoders and then applies near-anchor DualPoolFusion to selectively incorporate far-channel information through a sample-wise scalar gate and residual update.

For anomaly scoring, RRDS and sample-wise adaptive weighting were added to KMeans-based cosine scoring to balance source and target domain performance in the final score. This adaptive scoring improved pAUC among the single-model systems, and the weighted ensemble achieved the best official score of 64.86% on the development-test set. These results suggest that selective near and far fusion and adaptive reference-based scoring are effective for dual-channel first-shot ASD under source and target domain shift.

5. REFERENCES

[1] I. Ubhayaratne, Y. Xiang, M. Pereira, and B. Rolfe, "An audio signal based model for condition monitoring of sheet metal stamping process," in *Proc. IEEE International Conference on Industrial Electronics and Applications* (*ICIEA*), 2015, pp. 117–122.

[2] Y. Tagawa, R. Maskeliūnas, and R. Damaševičius, "Acoustic anomaly detection of mechanical failures in noisy real-life factory environments," *Electronics*, vol. 10, no. 19, p. 2329, 2021.

[3] K. Dohi, D. Niizumi, N. Harada, Y. Imoto, Y. Koizumi, T. Nishida, K. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2022 challenge task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," in *Proc. DCASE Workshop*, 2022, pp. 135–139.

[4] K. Dohi, Y. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, K. 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 Workshop*, 2023, pp. 31–35.

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

[6] T. Nishida, N. Harada, D. Niizumi, D. Albertini, R. Sannino, S. Pradolini, F. Augusti, Y. Imoto, K. Dohi, K. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2026 challenge task 2: Noise-aware unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE Workshop*, 2026.

[7] M. J. Alam, P. Ouellet, P. Kenny, and D. O'Shaughnessy, "Comparative evaluation of feature normalization techniques for speaker verification," in *Proc. International Conference on Nonlinear Speech Processing*, ser. Lecture Notes in Computer Science, vol. 7015. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 246–253.

[8] B. K. Choi, S. M. Ban, and H. S. Kim, "Cepstral feature normalization methods using pole filtering and scale normalization for robust speech recognition," *The Journal of the Acoustical Society of Korea*, vol. 34, no. 4, pp. 316–320, 2015.

[9] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, pp. 133–147, 1998.

[10] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "EAT: Self-supervised pre-training with efficient audio transformer," in *Proc. International Joint Conference on Artificial Intelligence* (*IJCAI*), 2024.

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

[12] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay attention to MLPs," *arXiv preprint arXiv:2105.08050*, 2021.

[13] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2019, pp. 4690–4699.

[14] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. European Conference on Computer Vision* (*ECCV*), 2016, pp. 499–515.

[15] J. Yang, "A two-stage fusion anomaly detection approach for DCASE 2025 task 2," DCASE 2025 Challenge Technical Report, 2025.

[16] 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 Workshop*, Barcelona, Spain, 2021, pp. 1–5.

[17] 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 Workshop*, Nancy, France, 2022.

[18] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. International Conference on Learning Representations* (*ICLR*), 2017.