ADAPTIVE MULTI-PARADIGM ENSEMBLE FOR NOISE-AWARE UNSUPERVISED ANOMALOUS SOUND DETECTION

Technical Report

Yu-jin Choi, Ji-sang Yoo, Dong-ha Oh, Seong-min Lee,
Jin-yang Lee, Hee-seok Jeon, Jung-hoon Noh

School of Semiconductor Engineering, Chungbuk National University, Cheongju, Republic of Korea
Corresponding author: jh.noh@cbnu.ac.kr

ABSTRACT

We present an adaptive multi-paradigm ensemble for noise-aware unsupervised anomalous sound detection. Our system combines eight orthogonal scorers—autoencoder (AE) reconstruction with Mahalanobis distance, BEATs-frozen k-NN, BEATs+LoRA discriminative k-NN, PaDiM-on-BEATs patch-wise density, Modulation-Spectrum Mahalanobis, Cross-channel Coherence, and two Masked-Spectrogram-Modeling (MSM) variants (random pixel mask and curriculum frame-block mask)—each capturing a distinct aspect of "normal-machine-sound" semantics. A novel *adaptive* λ rule maps training-data-only signal features (Hilbert-envelope kurtosis, Ch0–Ch1 Pearson correlation, attribute-CSV diversity) to per-machine ensemble weights, eliminating reliance on test-set labels at deployment time. On the DCASE 2026 Task 2 development set the submitted system achieves an official-score h-mean of **0.6300** (vs. 0.5789 baseline, $+0.051$ absolute). All score normalization uses domain-wise rank to neutralize source/target scale gaps.

***Index Terms*—** anomalous sound detection, domain generalization, ensemble learning, self-supervised learning, BEATs

1. INTRODUCTION

Anomalous sound detection (ASD) is the task of determining whether the sound emitted by a target machine is normal or anomalous. Anomalies in real-world factories are rare and diverse, which makes it infeasible to collect exhaustive patterns of anomalous sounds in advance. Therefore, an ASD system must learn from normal operating sounds alone and detect anomaly types unseen during training, a setting referred to as unsupervised ASD. The system must also remain reliable under domain shift, which arises from changes in the machine's operating conditions or surrounding environment.

DCASE 2026 Task 2 [1] imposes five requirements to address these real-world challenges. Four of them are retained from previous editions:

1. Unsupervised training using only normal sounds.
2. Domain generalization to handle domain shift.
3. Operation on new machine types unseen in development without additional hyperparameter tuning.
4. Operation both with and without attribute information.

The fifth requirement, introduced for the first time in DCASE 2026, is the use of two-channel audio recorded at different distances from the target machine: a near-mic (Ch0) and a far-mic (Ch1) that captures stronger environmental noise. Because the machine types in the evaluation set are completely different from those in the development set, manual tuning on the evaluation data is not possible.

The task is primarily concerned with building noise-robust systems that exploit the far-mic channel. Rather than targeting this new channel directly, our system focuses on maximizing detection performance by robustly addressing the core baseline requirements through a novel ensemble approach. It ensembles a diverse set of existing models from different paradigms, including a reconstruction-based autoencoder (AE) and pre-trained encoders, and combines their anomaly scores. Whereas z-score normalization is commonly applied to anomaly scores, we adopt domain-wise rank normalization, which is robust to outliers in the score distribution.

The ensemble weights are derived solely from signal features computed on each machine's normal recordings. Per-machine weights are assigned by a rule-based method according to features such as impulsiveness and inter-channel coherence. Because these features require no labels, the method can be applied to evaluation machines unseen during training without any label access. Specifically, our current system directly utilizes the near-mic channel (Ch0) as the primary source, as it captures the target signal more clearly. In addition, the far-mic channel (Ch1) is utilized complementarily by analyzing the relationship between Ch0 and Ch1 through the Cross-channel Coherence scorer.

2. PROPOSED METHOD

The proposed system comprises eight scorers, each capturing a different aspect of normal machine sound. The anomaly score of each scorer is normalized by domain rank and then combined through a weighted sum whose weights are set by an adaptive λ rule derived solely from training data (Fig. 1).

2.1. Individual Scorers

The scorers were selected to address specific weaknesses of the ensemble rather than combined arbitrarily. Starting from the AE baseline, each new scorer was added only if it improved a machine type where the existing scorers failed to detect anomalies, and only if it provided complementary, non-redundant information. This selection follows two primary design directions:

- **BEATs-based Scorers:** BEATs is a pre-trained model utilizing general audio (AudioSet), not machine operating sounds.

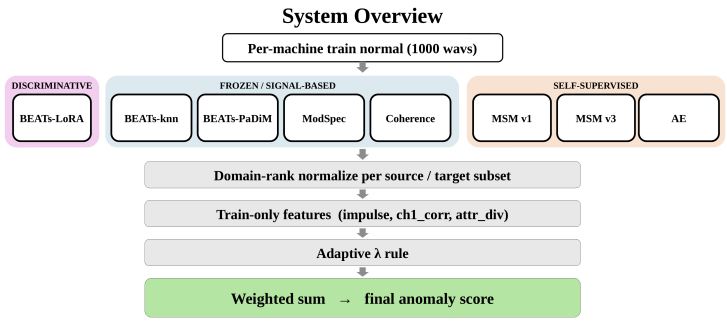

Figure 1: System overview: eight scorers, domain-rank normalization, and a train-only adaptive λ rule.

Because it was trained on millions of diverse environmental sounds, it captures rich acoustic structures that a machine-only autoencoder cannot capture. We leverage this single network as a frozen feature extractor in three distinct ways to yield different analytical views: the frozen k-NN uses the general representation directly, LoRA+SCAC adapts it to discriminate machine-specific operating conditions, and PaDiM models per-timestep distributions to capture localized deviations.

- **Signal-based Scorers:** ModSpec and Coherence contain no neural networks. Derived from classical vibration analysis [2], they encode physically defined fault signatures—namely impulsive modulation and near/far-mic coherence—which the neural scorers fail to model.

(a) Autoencoder (AE) + Mahalanobis. A fully-connected autoencoder following the DCASE first-shot baseline [3]. The input mel spectrogram (128 mels $\times$ 5 frames $=$ 640 dimensions) is compressed through four 128-dimensional layers to an 8-dimensional bottleneck and reconstructed by a symmetric decoder. It is trained per machine on source and target normal sounds. After training, per-domain covariances of the reconstruction errors give source and target Mahalanobis distances, and the test score is their minimum $\min(d_{\text{src}}, d_{\text{tgt}})$. This scorer measures how far the overall spectral statistics deviate from normal and serves as the base detector.

(b) BEATs-frozen k-NN. The frozen pre-trained BEATs encoder [4] produces a 768-d embedding by mean-pooling its final-layer output over time. The embedding is standardized and scored by cosine 1-NN distance to per-machine normal references. By relying on acoustic semantics learned from general audio, this scorer captures structure that the machine-only AE cannot.

(c) BEATs+LoRA+SCAC k-NN. The BEATs feature extractor is kept frozen, and only LoRA [5] adapters (rank=64) and a small projection head are trained. Training is performed jointly across all development machines with a Sub-cluster AdaCos [6] loss, so that the embedding adapts to discriminate machine-specific operating conditions. Standardized embeddings are scored by cosine 1-NN. This scorer discriminates categorical, attribute-related differences.

(d) PaDiM-on-BEATs. A multivariate Gaussian is fit per timestep over the BEATs patch tokens [7], using a random subset of channels and no learnable parameters. The per-timestep Mahalanobis distances are aggregated by their top-5 mean as the anomaly score. This scorer captures localized deviations where the distribution differs at specific time positions.

(e) Modulation Spectrum (ModSpec). This scorer uses no neural network. The Hilbert envelope of Ch0 is decimated to 1 kHz, a Welch-averaged power spectrum is computed over four frequency bands, and PCA followed by Mahalanobis distance is applied to per-machine statistics. It captures rotational periodicity and impulsiveness from classical vibration analysis [2], fault signatures that the neural scorers do not model.

(f) Cross-channel Coherence (Coh). This scorer also uses no neural network. The magnitude-squared coherence, cross-spectrum phase, and power-normalized cross-magnitude between Ch0 and Ch1 are computed, followed by PCA and Mahalanobis distance. It captures the stereo consistency between the two microphones, an invariant orthogonal to single-channel features, and is the only scorer that uses Ch1.

(g) MSM v1 (random-pixel mask). This scorer uses the same AE architecture with an added learnable mask token [8, 9]. During training, 75% of the input dimensions are randomly replaced by the mask token, and the MSE loss is computed only on the masked positions. Inference is unmasked, so the AE Mahalanobis pipeline is reused. Rather than memorizing normal spectra, the model learns to abstract normal structure by filling in the masked content.

(h) MSM v3 (curriculum frame-block mask). This scorer is identical to MSM v1 except that the masking unit is a whole mel frame and the mask ratio increases over training (2/5, then 3/5, then 4/5 of the frames). It learns longer-range temporal dependencies and captures a different aspect of normal structure than MSM v1.

2.2. Domain-Rank Normalization

All raw scores are normalized per-domain (source / target) by rank:

$$s_{\text{norm}}[i] = \frac{\text{rank}\big(s_{\text{raw}}[i] \mid \text{domain subset}\big)}{|\text{domain subset}|}. \tag{1}$$

This normalization effectively neutralizes the scale gap between the source and target domains. This scale gap occurs because target normal sounds are under-represented during training, which often causes them to produce higher raw reconstruction errors than actual source anomalies.

2.3. Adaptive Ensemble Weights

For each machine, three train-only features are extracted (Table 1). Six rules are then evaluated in a fixed order, and the first matching rule selects an 8-dimensional λ vector summing to one (Table 2). To resolve overlapping conditions where rules are nested, the narrower and more specific rule is placed first to take higher priority. The selected λ weights the domain-rank-normalized score of

each scorer, and their weighted sum is the final anomaly score,

$$s_{\text{final}} = \sum_{k=1}^{8} \lambda_k \, s_{\text{norm}}^{(k)}, \tag{2}$$

where $s_{\text{norm}}^{(k)}$ is the domain-rank-normalized score of scorer k and λ_k is the weight assigned to it by the selected rule. Because the entries of λ sum to one, the final score remains in $[0, 1]$. For unseen evaluation machines, the same features are computed on the supplemental training data. If no specialized rule matches, the system falls back to a default rule that applies a fixed, pre-determined global weighting optimized via grid search.

Crucially, this rule layer assigns weights based entirely on signal feature values rather than explicit machine types or labels. Consequently, the weight-assignment process depends solely on statistics derived directly from each machine's own normal recordings, requiring no labels and no test-set information. This allows the pre-determined weighting rules to be applied seamlessly to entirely new evaluation machines that were never encountered during the development phase. Whether a given evaluation machine receives a highly optimized specialized weighting depends on whether its extracted features fall within the calibrated rule ranges.

The three features utilized are complementary:

- The `impulse` feature separates impact-driven from steady-state machines.
- `ch1_corr` flags tightly coupled channels.
- `attr_div` tracks domain-shift severity.

Table 1: Train-only features for adaptive weighting.

Feature	Definition	Predicts
`impulse`	mean kurtosis of Ch0 Hilbert env.	impulsive character
`ch1_corr`	mean Pearson r of Ch0 vs Ch1	stereo coupling
`attr_div`	unique attr. combos in CSV	domain-shift complexity

Table 2: Adaptive λ rules: each column is the weighting calibrated for one development machine (columns sum to 1), selected by the train-only features of Table 1 so unseen evaluation machines match by features rather than identity. Rules apply in the listed priority order; DEF is the fallback for unmatched machines.

Scorer	Valve	Bearing	Fan	Gear	TCarEmu	Slider	Def
AE	0.00	0.10	0.15	0.20	0.55	0.40	0.30
BEATs-fz	0.05	0.00	0.10	0.00	0.05	0.05	0.10
LoRA	0.10	0.00	0.15	0.60	0.10	0.15	0.25
PaDiM	0.00	0.00	0.35	0.05	0.25	0.35	0.25
ModSpec	0.55	0.00	0.15	0.10	0.05	0.05	0.10
Coh	0.30	0.00	0.10	0.05	0.00	0.00	0.00
MSM v1	0.00	0.40	0.00	0.00	0.00	0.00	0.00
MSM v3	0.00	0.50	0.00	0.00	0.00	0.00	0.00

Triggers (priority order): Valve `imp` >8; Bearing `imp` >3; Fan `ch1` >0.7; Gear `attr` $\in [5, 12]$ & $|$`ch1`$|<0.3$; TCarEmu `attr` >12; Slider `ch1` $\in (0.3, 0.6)$ & `attr` ≤ 1 & `imp` <1; Def otherwise.

3. EXPERIMENTAL SETUP

3.1. Datasets

The development set consists of seven machine types used for system design and λ weight fitting. These machine types are drawn from the ToyADMOS2 [10] and MIMII DG [11] datasets. For each machine, the training data provides 990 source-normal and 10 target-normal audio clips. The test data provides 50 clips each for the combination of source/target and normal/anomaly conditions, with ground-truth labels provided. Additional training data serves as a supplemental set, featuring five new machine types with 990 source-normal and 10 target-normal clips each. The evaluation set for the final submission targets the same five machine types from the supplemental set, containing 200 unlabeled test clips each. All audio data is recorded as 16 kHz stereo, with a clip length of 10 s for most machines, 6 s for ToothBrush, and 16 s for ToyDrone.

3.2. Training Configuration

We employ the Adam optimizer with a learning rate of 10^{-3} and no weight decay for the Autoencoder, MSM v1, and MSM v3. For BEATs+LoRA, we use the AdamW optimizer with a learning rate of 10^{-4} and a weight decay of 10^{-4}, accompanied by a 5000-step warmup phase. The batch size is set to 256, except for the 16-s ToyDrone clips which use a batch size of 64. Training runs for 100 epochs per machine, followed by a single additional epoch dedicated to covariance estimation.

The input Mel spectrogram configuration consists of 128 mels, an FFT size of 1024, a hop length of 512, and a 5-frame concatenation that yields a 640-dimensional vector. All scorers process the near-mic channel (Ch0) exclusively, with the sole exception of the Coherence scorer which utilizes both Ch0 and Ch1. We index the near-mic as Ch0 and the far-mic as Ch1, which directly correspond to "channel 1" and "channel 2" defined in the official challenge. The random seed is fixed to 13711 for reproducibility.

4. RESULTS

4.1. Development-set h-mean Progression

The adaptive λ rule layer is the main contributor to performance optimization (Table 3): it increases the six-model global score from 0.6033 to the submitted eight-scorer result of 0.6300 ($+0.027$). The per-machine oracle (0.6563) marks the upper bound reachable only if eval-time weights were available—which the rule design deliberately avoids depending on.

Notably, the global ensemble performance remains limited when Coherence is added on top of ModSpec ($0.6041 \rightarrow 0.6033$). Under a single global λ, the large weights assigned to specialized machines are diluted across the other machines where they do not improve detection accuracy. The adaptive layer resolves this by granting ModSpec and Coherence weight only where the `impulse` feature exceeds the activation threshold, converting the 0.6033 score into 0.6300.

Table 3: Development-set h-mean progression.

System	h-mean	vs base
AE baseline (MAHALA)	0.5789	—
4-model Global	0.6013	$+0.022$
5-model Global $+$ModSpec	0.6041	$+0.025$
6-model Global $+$Coh	0.6033	$+0.024$
Adaptive λ+MSM (8 mdl)—submitted	**0.6300**	$+0.051$
Per-machine oracle (upper bound)	0.6563	$+0.077$

4.2. Per-machine Breakdown

Seven development machines were evaluated (Table 4), and five of them meet or exceed the AE baseline. The evaluation confirms

Table 4: Per-machine breakdown of the submitted system (domain-rank-normalized, adaptive λ).

Machine	AUC(src)	AUC(tgt)	pAUC	h	vs base h	Active rule
ToyCar	0.631	0.688	0.523	0.6061	-0.003	default
ToyCarEmu	0.617	0.739	0.541	0.6220	-0.020	Rule 5
bearingEmu	0.703	0.617	0.594	0.6343	$+0.013$	Rule 2 (MSM v1+v3)
fan	0.571	0.509	0.542	0.5393	$+0.010$	Rule 3
gearboxEmu	0.755	0.700	0.623	0.6883	$+0.101$	Rule 4 (LoRA)
sliderEmu	0.598	0.557	0.531	0.5605	$+0.023$	Rule 6
valveEmu	0.912	0.874	0.761	**0.8439**	$+0.298$	Rule 1 (ModSpec+Coh)

that each model paradigm contributes by capturing a distinct feature of the signal, which the routing layer then allocates to the corresponding machine types.

The signal-based ModSpec and Coherence effectively encode the impulsive vibration patterns of impact mechanics. This classical condition-monitoring approach captures features that the mel-AE cannot represent, effectively improving the detection performance of the impulsive valveEmu by $+0.298$. The attribute-discriminative BEATs+LoRA embedding captures domain-invariant acoustic features. While the machine-only AE performance drops under domain shift, this large-scale pre-trained model ensures robust generalization and improves gearboxEmu by $+0.101$.

PaDiM captures localized per-timestep deviations and optimizes performance on fan types routed to it by their high inter-channel coherence (`ch1_corr` $= 0.87$, $+0.010$). The two masked-spectrogram variants learn an abstracted normal structure, uniquely improving performance on impulsive bearingEmu ($+0.013$) under specialized routing. In contrast, the vehicle-type variants (such as ToyCar and ToyCarEmu) fell slightly below the baseline. Their high attribute diversity either saturated the discriminative capacity of the LoRA head or offered no specialized signature for the rule layer to exploit, leaving them on the default weighting.

This allocation to the default weighting highlights a key limitation of our current hard thresholding mechanism, which is calibrated across only seven development machine types. If an unseen machine's signal features slightly miss these sharp activation thresholds, the discrete approach fails to allocate optimized weights. To address this issue, future work will focus on extending the rule layer into a continuous framework. Rather than relying on six discrete rules, the system can shift to a continuous interpolation method, where ensemble weights are formulated as continuous functions of signal features, such as a sigmoidal mapping. Furthermore, we plan to calibrate the system using a wider variety of models and datasets.

5. CONCLUSION

We propose an adaptive multi-paradigm ensemble for unsupervised ASD under domain shift, which extracts train-only signal features to dynamically route weight assignments without relying on explicit machine labels. The adaptive routing is the core method of our system: by allocating specific scorer configurations based on measured signal characteristics, it increases the global ensemble score from 0.6033 to the submitted 0.6300 ($+0.051$ over the official AE baseline). The per-machine evaluation confirms that matching model paradigms to signal types drives overall detection accuracy: signal-based ModSpec and Coherence improve performance on the impulsive valveEmu ($+0.298$), while the attribute-discriminative BEATs+LoRA embedding optimizes performance on gearboxEmu ($+0.101$).

The masked-spectrogram scorers operate conditionally within this framework. They contribute to performance on impulsive bearings on the development set, but remain inactive on the five evaluation machines because the measured impulse values (-1.3 to 2.4) do not meet the activation threshold. However, the adaptive rule layer successfully identified this change in signal characteristics and assigned a weight of zero to the masked-spectrogram scorers. This mechanism allowed the system to fall back to the 0.6041 default global weighting, maintaining baseline robustness on the evaluation set.

6. REFERENCES

[1] T. Nishida *et al.*, "Description and discussion on DCASE 2026 Challenge Task 2: Noise-aware unsupervised anomalous sound detection for machine condition monitoring," arXiv:2606.01578, 2026.

[2] R. B. Randall, *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*. Wiley, 2011.

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

[4] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proc. ICML*, 2023.

[5] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022.

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

[7] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: a patch distribution modeling framework for anomaly detection and localization," in *Proc. ICPR Int. Workshops and Challenges*, 2021, pp. 475–489.

[8] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. CVPR*, 2022.

[9] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," in *Proc. NeurIPS*, 2022.

[10] 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*, 2021, pp. 1–5.

[11] 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*, 2022.