# NOISE-AWARE UNSUPERVISED MACHINE ANOMALOUS SOUND DETECTION USING PRE-TRAINED ACOUSTIC REPRESENTATIONS, DUAL-CHANNEL GATED FUSION, AND NORMAL-REFERENCE MODELING

*Technical Report*

Lianzhi Wang, Jeffrey Liu

*Suzhou Dongyuan Electronics Co., Ltd., China*

Lianzhi Wang, wanglz@dongyuansz.com

This report describes our system for DCASE 2026 Challenge Task 2, noise-aware unsupervised anomalous sound detection for machine condition monitoring. The task requires a system to use only normal training clips and to output continuous anomaly scores for unlabeled clips in the evaluation dataset; auxiliary decision files are generated only to match the submission package format. The system must address source-target domain shift, two-channel noisy recordings, a very small number of target-domain normal clips, and the generalization risk caused by the mismatch between development and evaluation machine types. We use pre-trained acoustic representation models as audio encoders and adapt them with Low-Rank Adaptation (LoRA). The main branch is based on BEATs, while the auxiliary branch is based on AudioMAE. Both branches use a dual-channel gated fusion module to map synchronized near-field and far-field channels into sample-level embeddings. During training, ArcFace-style attribute and domain classification objectives are used to shape the embedding space of normal samples. During inference, the classification heads are removed, normal training samples are used to build reference banks, and anomaly scores are computed by distance-based normal-reference scoring. Because the raw anomaly scores produced by different scoring backends have different scales, we calibrate the scores from the two branches by robust z-score calibration estimated only from normal reference scores, and then perform score-level fusion. On the development dataset, the BEATs-AudioMAE robust-z weighted ensemble with BEATs weight 0.85 and left-channel-duplicated inference obtains an overall harmonic mean of 61.43%. This result is higher than the official simple autoencoder baseline of 56.66% and the selective Mahalanobis baseline of 57.66%. To reduce overfitting risk on the hidden evaluation set, we also submit a default two-channel inference variant, a more conservative fusion variant with a higher BEATs weight, and a BEATs single-encoder variant.

Index Terms: anomalous sound detection, machine condition monitoring, pre-trained audio representation, BEATs, AudioMAE, normal reference modeling

## 1. INTRODUCTION

Unsupervised anomalous sound detection (ASD) aims to identify whether a test signal deviates from normal operation when no anomalous training examples are available. This setting is important for industrial machine monitoring, where real anomalous examples are often rare, costly to collect, and open-ended in type. The DCASE Task 2 series further introduces machine type, section, attribute, source/target domain, and first-shot adaptation factors. Therefore, a system must not only model the distribution of normal sounds, but also generalize across machine types and recording conditions.

DCASE 2026 Task 2 extends the previous first-shot unsupervised ASD setting with a stronger noise-aware two-channel recording condition. Each audio clip contains two synchronized channels, which usually correspond to different microphone distances or different noise observation conditions. The near-field channel often has a higher signal-to-noise ratio for the target machine, whereas the far-field channel may contain more environmental noise, room response, and domain-shift information. Therefore, directly fixing one channel may discard useful complementary information, while simple concatenation or averaging may introduce noise. We use a learnable gated fusion mechanism to learn soft channel selection during training, and retain both default two-channel inference and left-channel-duplicated inference as submitted variants.

Recent pre-trained acoustic representation models have shown strong generalization ability in small-sample ASD. BEATs, AudioMAE, EAT, PANNs, and DaSheng learn general acoustic priors from large-scale audio corpora, and often outperform shallow models or autoencoders trained from scratch after modest adaptation to machine sounds. Our system follows a “pre-trained

acoustic encoder + discriminative representation learning + normal-reference scoring" framework. During training, attribute and domain labels of normal samples provide supervision. During inference, test embeddings are compared only with normal reference embeddings.

The main contributions of this system are threefold. First, we build a discriminative ASD framework based on pre-trained acoustic representations, transfer the general audio priors of BEATs and AudioMAE to normal machine-sound modeling, and use LoRA to reduce overfitting in small-sample adaptation. Second, we design a dual-channel gated fusion strategy and ArcFace auxiliary representation-learning objectives, so that normal embeddings preserve machine-attribute, domain, and channel-complementary information. Third, we use normal-reference distribution modeling for anomaly scoring, and perform calibration only from normal training reference scores, without using the score distribution of the evaluation test set.

## 2. TASK SETTING AND DATA USAGE

The task is an unsupervised anomaly detection task. During training, only normal clips are available, and the labels of evaluation test clips are hidden. The core system output is a continuous anomaly score for each test clip, which indicates the degree of deviation from the normal reference distribution and is used for AUC and pAUC computation. Binary decision files are generated only as auxiliary submission-format files and are not used for score aggregation or model selection.

Data usage is divided into two stages. The first stage is development model selection. We train candidate systems on the development dataset and use the development test labels to compute AUC(source), AUC(target), pAUC, and the official harmonic mean. These results are used to select the backbone, checkpoint, embedding dimension, fusion setting, and scoring backend. The second stage is final adaptation. For the final submission, the models are adapted using the normal clips from the additional training dataset. The same normal clips are also used to build normal reference banks for the evaluation machines. Test clips in the Evaluation Dataset are used only for per-clip embedding extraction and anomaly-score output; they are not used for training, model selection, or score calibration.

This usage follows the unsupervised setting of Task 2. The additional training dataset is the official normal training data, and the Evaluation Dataset is the test data to be scored. We do not use evaluation test labels, and we do not use the global score distribution of evaluation test clips for rank normalization or score calibration. All score-calibration statistics are derived from accessible normal reference scores.

## 3. SYSTEM METHOD

### 3.1 Overall Pipeline

The system input is a two-channel waveform. Each channel is first resampled and cropped or zero-padded to a fixed duration. The two channels are separately fed into a shared acoustic encoder to obtain token-level or frame-level representations. The two-channel representations are then combined by a learnable gated fusion module into a single sample-level representation, and a projection head maps it to a 256-dimensional embedding. During training, this embedding is passed to ArcFace attribute and domain classification heads. During inference, the classification heads are removed, and only the embedding extractor and anomaly scoring backend are retained.

For a single-model system, the test embedding is directly compared with the normal reference bank of the same machine type to obtain the anomaly score. For a dual-encoder system, the anomaly scores from the two encoders are first computed from their respective normal reference distributions, and then calibrated and weighted during final submission.

### 3.2 Pre-trained Acoustic Representations

We do not train the acoustic front end from scratch. Instead, large-scale audio pre-trained models are used as feature extractors. BEATs learns discrete acoustic-token representations through acoustic tokenization and masked prediction, and provides strong discrimination for broad audio events. AudioMAE learns time-frequency patch-level representations through masked autoencoding, emphasizing the reconstruction of complete acoustic structure from partial observations. Because their pre-training objectives are different, they may have different sensitivity to stationary harmonics, transient impacts, and background noise in machine sounds. We use BEATs as the main encoder because it is the most stable model on the development dataset, and use AudioMAE as an auxiliary encoder to provide complementary representations from a different pre-training objective.

Given a two-channel waveform $x = [x^{(1)}, x^{(2)}]$, each channel is passed through the same pre-trained encoder $f_\theta$ to obtain frame-level representations:

$$H^{(c)} = f_\theta(x^{(c)}), \quad c \in \{1, 2\}.$$

The system then obtains a sample-level embedding $e \in \mathbb{R}^{256}$ through channel fusion, temporal pooling, and a projection head. This embedding is the common input to ArcFace representation learning and normal-reference distribution modeling. Since final anomaly detection depends on distances between embeddings and normal reference banks, the goal of the acoustic encoder is not to classify anomalies directly, but to construct an embedding space in which normal machine sounds are compact within internal structures and separable across attributes and domains.

### 3.3 LoRA Adaptation

Let $W$ denote a pre-trained linear-layer weight. LoRA represents the task-specific update as a low-rank matrix product:

$$W' = W + \Delta W, \quad \Delta W = \frac{\alpha}{r} BA,$$

where $r$ is the rank, $\alpha$ is the scaling factor, and $A$ and $B$ are trainable low-rank matrices. We use LoRA $r = 8$, $\alpha = 16$, and dropout $= 0.1$. Most pre-trained parameters of BEATs and AudioMAE are frozen. The LoRA adapters, classification heads, and fusion module are trainable. This strategy substantially reduces the number of trainable parameters and makes the model more suitable for target-machine adaptation with only normal samples.

Compared with full fine-tuning, LoRA adaptation has two advantages. First, it limits the degrees of freedom of task-specific updates, making the model less likely to memorize recording conditions from the small number of target-domain normal clips. Second, the frozen pre-trained parameters preserve general acoustic priors learned from large-scale audio corpora. For Task 2, where the evaluation machines differ from the development machines, parameter-efficient adaptation is more appropriate than full retraining as a final submission strategy.

### 3.4 Dual-channel Gated Fusion

Let $h_t^{(1)}$ and $h_t^{(2)}$ be the token representations of the two channels after the pre-trained encoder, where channel 1 is treated as the near-field channel and channel 2 as the far-field channel. The gated fusion module first estimates a channel weight from the concatenated representations:

$$\alpha_t = \sigma(W_g[h_t^{(1)};h_t^{(2)}] + b_g),$$

and then computes the fused representation:

$$h_t = \alpha_t h_t^{(1)} + (1 - \alpha_t) h_t^{(2)}.$$

Here, $\sigma(\cdot)$ denotes the sigmoid function. The gate bias is initialized to 1.0, which makes the model moderately favor the near-field channel at the beginning of training while still preserving complementary information from the far-field channel. This design is more flexible than fixed channel selection and more robust than simple averaging, because the model can adaptively change the contribution of each channel at different time frames.

During training, channel swap, channel gain perturbation, and channel dropout are used as waveform-level augmentations. Channel swap reduces the dependence on a fixed channel order, channel gain perturbation simulates microphone gain differences, and channel dropout forces the model to produce stable embeddings under single-channel degradation. All final submitted models use default dual-channel training. Left-channel-duplicated inference is used only as an inference-stage channel-selection variant, and is not left-only training.

### 3.5 ArcFace Representation Learning

Two auxiliary classification tasks are used during training: attribute classification and domain classification. For an ArcFace classification head, let $\theta_y$ denote the angle between the normalized embedding and the normalized class weight of the ground-truth class. An angular margin $m$ is added to the ground-truth logit:

$$\ell_y = s\cos(\theta_y + m), \quad \ell_j = s\cos(\theta_j), j \neq y,$$

where $s$ is a scale factor. The total training objective is:

$$\mathcal{L} = \mathcal{L}_{attr} + \mathcal{L}_{domain}.$$

When a sample does not have a valid attribute label, $\mathcal{L}_{attr}$ is masked for that sample, while the sample still contributes to domain classification. The ArcFace objective is used only to learn a compact and separable embedding space for normal samples. During anomaly detection, the classification heads are not used, and anomalous samples are never used as supervised classes.

ArcFace is used instead of a standard softmax classifier because the anomaly scoring backend relies on distance relationships in the embedding space. The angular-margin constraint in ArcFace explicitly compresses normal samples of the same class and increases angular distances between different attributes or domains, which makes the local structure of normal reference banks clearer. Consequently, when a test sample deviates from normal attribute or normal domain structure, its distance-based anomaly score is more likely to increase.

## 4. ANOMALY SCORING AND SCORE CALIBRATION

### 4.1 BEATs Distance Scoring

The BEATs branch models the normal embedding distribution with Mahalanobis distance. For each normal reference bank, a Ledoit-Wolf covariance estimator is used to estimate the mean $\mu$ and inverse covariance matrix $\Sigma^{-1}$. The anomaly score of a test embedding $x$ is:

$$s_B(x) = \sqrt{(x - \mu)^\top \Sigma^{-1} (x - \mu)}.$$

In the domain-conditioned bank mode, source normal clips and target normal clips form separate reference banks. Since the domain label of evaluation test clips is hidden, the system does not choose a bank according to the test domain. Instead, it computes the anomaly score to all available banks and takes the minimum:

$$s_B(x) = \min_{b \in \mathcal{B}} s_b(x).$$

This rule is equivalent to selecting the normal bank that places the test sample closest to the normal distribution, without relying on hidden domain labels.

### 4.2 AudioMAE Nearest-neighbor Scoring

The AudioMAE branch uses non-parametric k-nearest-neighbor cosine-distance scoring. Let $\{b_i\}_{i=1}^{N}$ denote the normal reference embeddings, $x$ the test embedding, and $d(\cdot,\cdot)$ the cosine distance. If $\mathcal{N}_k(x)$ denotes the indices of the $k$ nearest normal samples to $x$, the anomaly score is:

$$s_A(x) = \frac{1}{k}\sum_{i \in \mathcal{N}_k(x)} d(x, b_i)\,.$$

The AudioMAE branch uses $k = 3$, a global normal reference bank, and raw distance scores. This backend makes weaker parametric assumptions about the normal distribution and serves as a complementary branch to BEATs Mahalanobis scoring.

## 5. DEVELOPMENT RESULTS

Table 1 reports per-machine results and overall harmonic means on the development dataset. All scores are percentages. Official baseline numbers are taken from the task description for the simple autoencoder mode and the selective Mahalanobis mode. Baseline standard deviations are omitted here, and only mean values are retained for comparison with the submitted systems.

Table 1: AUC and pAUC results on the development dataset

| Machine | Metric | Baseline MSE | Baseline Mahalanobis | task2_1 | task2_2 | task2_3 | task2_4 |
|---|---|---|---|---|---|---|---|
| bearingEmu | AUC(source) | 62.34 | 65.92 | 56.36 | 57.32 | 56.76 | 56.88 |
| | AUC(target) | 59.56 | 62.28 | 58.84 | 55.44 | 58.80 | 58.76 |
| | pAUC | 59.85 | 60.42 | 55.47 | 53.37 | 55.37 | 55.47 |
| fan | AUC(source) | 61.45 | 60.00 | 70.80 | 75.76 | 70.20 | 68.20 |
| | AUC(target) | 46.94 | 45.09 | 57.92 | 60.80 | 56.48 | 54.32 |
| | pAUC | 53.33 | 52.29 | 52.37 | 53.16 | 52.16 | 52.00 |
| gearboxEmu | AUC(source) | 68.23 | 74.48 | 77.16 | 77.04 | 76.52 | 75.44 |
| | AUC(target) | 49.78 | 52.74 | 57.88 | 58.36 | 57.60 | 57.36 |
| | pAUC | 52.94 | 53.97 | 53.21 | 52.95 | 53.11 | 52.89 |
| sliderEmu | AUC(source) | 67.25 | 66.36 | 49.48 | 48.84 | 49.16 | 49.20 |
| | AUC(target) | 45.05 | 49.18 | 55.88 | 55.52 | 56.16 | 56.76 |
| | pAUC | 50.38 | 50.36 | 48.79 | 48.16 | 48.79 | 48.89 |
| ToyCar | AUC(source) | 75.62 | 77.28 | 73.16 | 74.36 | 73.00 | 72.88 |
| | AUC(target) | 37.87 | 53.17 | 81.36 | 80.60 | 81.24 | 80.80 |
| | pAUC | 54.03 | 58.25 | 63.68 | 63.05 | 63.58 | 63.00 |
| ToyCarEmu | AUC(source) | 69.62 | 69.49 | 70.32 | 68.64 | 71.36 | 72.32 |
| | AUC(target) | 61.20 | 66.62 | 69.32 | 68.28 | 67.48 | 65.00 |
| | pAUC | 55.89 | 53.47 | 55.26 | 54.68 | 55.95 | 57.53 |
| valveEmu | AUC(source) | 67.74 | 56.60 | 89.56 | 90.12 | 89.40 | 89.48 |
| | AUC(target) | 68.78 | 56.50 | 68.60 | 67.24 | 68.56 | 68.72 |
| | pAUC | 55.08 | 50.20 | 59.37 | 60.26 | 59.42 | 59.47 |
| All (hmean) | AUC(source) | 67.18 | 66.46 | 67.25 | 67.82 | 67.20 | 66.95 |
| | AUC(target) | 50.85 | 54.24 | 63.22 | 62.75 | 62.73 | 62.06 |
| | pAUC | 54.37 | 53.91 | 55.10 | 54.71 | 55.12 | 55.25 |

Compared with the official baselines, the main improvement of the four submitted systems comes from target-domain AUC. The selective Mahalanobis baseline has an AUC(target) harmonic mean of 54.24%, whereas all four submitted systems exceed 62.0%. The improvement in pAUC is smaller, indicating that the low false-positive-rate region remains the main direction for further optimization.

## 6. SUBMITTED SYSTEMS

We submit four systems. `task2_1`, `task2_2`, and `task2_3` use a dual-encoder combination of BEATs and AudioMAE with score-level weighted aggregation. They mainly differ in inference-channel selection and the BEATs branch weight. `task2_4` uses a single BEATs encoder with Mahalanobis scoring and is included as a simpler single-model submission. All four systems use default dual-channel training and gated two-channel fusion

during training. Left-channel duplication is used only during inference for selected submitted systems, and is not left-only training.

## 7. COMPLIANCE AND EXTERNAL RESOURCES

The system uses only the official development dataset, additional training dataset, and Evaluation Dataset. Development labels are used only for model selection and result reporting during the development phase. In the final submission phase, we do not use Evaluation Dataset labels, and we do not use the overall score distribution of the Evaluation Dataset for model selection or score calibration.

Regarding external resources, the system uses the official BEATs pre-trained checkpoint and an AudioMAE pre-trained model. Both are disclosed in the metadata as pre-trained acoustic representation resources.

## 8. CONCLUSION

We presented a DCASE 2026 Task 2 system based on pre-trained acoustic representations, dual-channel gated fusion, ArcFace representation learning, and normal-reference scoring. The BEATs single model provides stable base performance, while AudioMAE provides a complementary acoustic representation learned from a different pre-training objective. On the development dataset, the best submitted system, `task2_1`, obtains an overall harmonic mean of 61.43%, outperforming the official simple autoencoder baseline of 56.66% and the selective Mahalanobis baseline of 57.66%. The final four submitted systems share the same core modeling framework and differ only in channel selection, encoder combination, and reference-score aggregation, while avoiding any post-processing tuned from the evaluation test distribution.

## REFERENCES


[1] Tomoya Nishida, Noboru Harada, Daiki Takeuchi, Daisuke Niizumi, Keisuke Imoto, Kota Dohi, Harsh Purohit, Takashi Endo, and Yohei Kawaguchi. Description and discussion on DCASE 2026 Challenge Task 2: Noise-aware unsupervised anomalous sound detection for machine condition monitoring. arXiv e-prints, 2026.

[2] Takuya Fujimura, Ibuki Kuroyanagi, and Tomoki Toda. The NU Systems for DCASE 2025 Challenge Task 2. DCASE 2025 Challenge Technical Report, 2025.

[3] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. BEATs: Audio Pre-Training with Acoustic Tokenizers. Proceedings of the 40th International Conference on Machine Learning, 2023.

[4] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked Autoencoders that Listen. Advances in Neural Information Processing Systems, 2022.

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. International Conference on Learning Representations, 2022.

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

[7] Ross Wightman. PyTorch Image Models. https://github.com/huggingface/pytorch-image-models.