DISCRIMINATIVE RESNET AND BEATS ADAPTATION FOR NOISE-AWARE MACHINE SOUND ANOMALY DETECTION

Technical Report

Jiakun Xia

Northeastern University, China
whfdyer@163.com

ABSTRACT

DCASE2026 Task2 evaluates first-shot unsupervised anomalous sound detection under unseen machine types and noisy two-channel recordings. We submit four systems that share a common training, embedding, and scoring pipeline: a trainable MultiResNet spectral encoder, two adapted BEATs encoders, and a BEATs LoRA adaptation system. The first system can use channel 2 as a deterministic far-field noise reference during inference preprocessing, while the three BEATs systems are submitted with the raw channel-1 inference view. All systems use sub-cluster AdaCos supervision with 16 subclusters and cosine nearest-neighbor scoring against normal training embeddings.

***Index Terms*—** anomalous sound detection, machine condition monitoring, pretrained audio models, ResNet, Wiener filtering

1. INTRODUCTION

Anomalous sound detection (ASD) for machine condition monitoring aims to identify abnormal machine sounds when only normal recordings are available for training. DCASE2026 Task2 combines first-shot machine generalization with a noise-aware two-channel recording setup [1, 2], building on ToyADMOS2, MIMII DG, and first-shot ASD benchmarks [3, 4, 5]. A submitted system must therefore produce anomaly scores for hidden machine types using only the released training recordings and the test recordings at inference time.

Our submission is implemented in an ASDKit-style four-stage pipeline of labeling, training, feature extraction, and scoring [6]. The four submitted systems use the same metadata-derived labels, embedding objective, and backend. They differ in frontend representation and adaptation mechanism: one system learns a MultiResNet spectral encoder, two systems adapt BEATs with different discriminative fine-tuning settings, and the fourth uses BEATs with LoRA adaptation [7, 12].

We reuse substantial implementation components from ASDKit, including its public training, extraction, and scoring framework. Our submission-specific choices are the selected four-system slate, the adaptation settings, and the deterministic channel-2-aware preprocessing choices used for the DCASE2026 submission.

Each frontend is trained on channel-1 audio with sub-cluster AdaCos [8]. At inference time, normal training embeddings define the machine reference set, and anomaly scores are computed with cosine nearest-neighbor distance. Larger scores indicate greater distance from the normal reference set and therefore higher anomaly likelihood.

2. PROPOSED SYSTEMS

2.1. Common training and scoring pipeline

All submitted models are trained on channel-1 recordings from the DCASE2026 training data. Training follows the official normal-only setting but converts available machine and section metadata into a discriminative proxy-label objective. The embedding extractor is optimized with sub-cluster AdaCos using 16 subclusters per class. Mixup with probability 0.5 is applied during frontend training [9].

At inference time, embeddings are L2-normalized before scoring. The primary backend is a cosine k-nearest-neighbor scorer that queries one neighbor from the source-domain reference pool and one neighbor from the target-domain reference pool. Before fitting the target-domain index, we apply light SMOTE augmentation with a 10% oversampling ratio and two neighbors. The anomaly score is the nearest-neighbor distance, with larger values indicating higher anomaly likelihood.

The backend is fitted separately from the frontend training step. For each machine section, normal training embeddings are divided by domain metadata into source and target reference pools. With cosine scoring, embeddings are first normalized to unit length and then queried with a Euclidean nearest-neighbor index, which is equivalent to a monotonic cosine-distance score. When target-domain normal examples are available, the target reference pool is augmented by SMOTE before nearest-neighbor fitting [10]; at test time, source and target distances are computed separately and the smaller distance is used as the final anomaly score. This keeps the score direction consistent across all four systems while allowing the same backend to handle both source-like and target-like recordings.

2.2. MultiResNet spectral encoder

The first system uses the trainable MultiResNet frontend. Given a waveform, the frontend computes one full-clip FFT branch and three STFT branches with FFT sizes 256, 1024, and 4096, using hop sizes 128, 512, and 2048. The STFT branches use linear-frequency magnitude spectra from 200 Hz to 8 kHz. Each branch is encoded by a ResNet-style convolutional network [11] with squeeze-and-excitation blocks and mapped to a 128-dimensional branch embedding; the branch outputs are concatenated into the final clip embedding. The submitted System 1 output averages the raw, mild Wiener, and strong Wiener inference views of this frontend.

2.3. BEATs adaptation systems

Systems 2–4 use BEATs as a pretrained audio representation [7]. Each restores the BEATs Iter3 checkpoint, disables BEATs

Table 1: Development-set diagnostic scores (%).

Machine	Metric	System 1	System 2	System 3	System 4
bearingEmu	AUC_s	59.04	60.92	59.08	57.20
	AUC_t	54.76	50.84	51.68	56.40
	pAUC	49.78	55.53	55.83	54.34
	hmean	54.26	55.46	55.36	55.95
fan	AUC_s	71.92	62.08	64.28	59.16
	AUC_t	55.88	45.92	45.20	46.44
	pAUC	53.43	53.73	55.46	54.63
	hmean	59.39	53.10	53.85	52.87
gearboxEmu	AUC_s	74.00	72.80	74.28	74.32
	AUC_t	74.52	65.00	67.80	66.32
	pAUC	57.45	53.91	55.43	53.62
	hmean	67.66	62.93	64.86	63.58
sliderEmu	AUC_s	75.84	68.60	68.72	68.28
	AUC_t	75.52	56.56	57.80	58.40
	pAUC	55.60	51.15	50.90	49.76
	hmean	67.55	57.91	58.25	57.84
ToyCar	AUC_s	69.72	74.20	75.64	74.32
	AUC_t	76.76	83.56	80.32	81.04
	pAUC	57.29	57.52	53.29	55.19
	hmean	66.92	70.04	67.51	68.32
ToyCarEmu	AUC_s	65.84	58.84	58.48	56.88
	AUC_t	96.52	84.96	85.64	85.16
	pAUC	50.22	51.70	50.28	48.49
	hmean	65.99	62.36	61.65	60.06
valveEmu	AUC_s	98.60	96.24	96.36	97.92
	AUC_t	86.32	86.20	84.84	89.40
	pAUC	84.75	68.32	67.76	74.49
	hmean	89.48	81.90	81.25	86.15
hmean	AUC_s	72.01	68.83	69.17	67.48
	AUC_t	71.58	63.72	63.96	65.58
	pAUC	56.74	55.52	55.10	54.85
	hmean	65.96	62.20	62.19	62.11

SpecAugment during task adaptation, and trains a sub-cluster AdaCos embedding head for the machine-section proxy labels. System 2 uses low-learning-rate task adaptation with AdamW at 5×10^{-5} for 25 epochs. System 3 uses discriminative task fine-tuning for the same duration with a learning rate of 10^{-4}. System 4 uses BEATs LoRA adaptation with rank 128 and a 40-epoch training schedule [12].

For every submitted system, each score stream applies the same sequence of waveform preprocessing, clip-level embedding extraction, L2 normalization, and shared KNN scoring. The MultiResNet system changes the spectral representation and trainable ResNet encoder architecture. The BEATs systems keep the pretrained audio-tokenizer representation family fixed while changing the amount and form of task adaptation. The backend interface is the same for all four systems: it receives a matrix of normal training embeddings and a matrix of test embeddings, then returns one continuous anomaly score for each test clip.

2.4. Evaluation-time Wiener preprocessing

For Wiener-preprocessed score streams, the second channel is used only as a noise reference for preprocessing [13]. Let $X_1(f,t)$ and $X_2(f,t)$ be the STFTs of the near-field and far-field channels after RMS matching. The Wiener-style gain is

$$G(f,t) = \text{clip}\left(1 - \alpha \frac{|X_2(f,t)|^2}{|X_1(f,t)|^2 + \epsilon}, \beta, 1\right), \qquad (1)$$

and the frontend receives $\text{iSTFT}(G(f,t)X_1(f,t))$. The mild branch uses $\alpha = 0.5$ and $\beta = 0.05$, and the strong branch uses $\alpha = 1.0$ and $\beta = 0.01$. This step is a deterministic channel-2-aware transform before embedding extraction, not a separate trained model.

2.5. Output generation

For each submitted system, training embeddings are first extracted from the normal recordings and fitted by the shared backend. Test recordings are then passed through the selected preprocessing, frontend, L2 normalization, and KNN scorer. The submitted score files store one anomaly score per test clip and machine section. The required binary decision files are generated by a fixed per-machine threshold rule: the threshold is the 0.9 quantile of normal training scores and does not use the evaluation-test score distribution.

3. CONCLUSION

We submit four ASD systems for DCASE2026 Task2. The systems consist of a MultiResNet spectral encoder, two BEATs adaptation anchors, and a BEATs LoRA adaptation system. All systems use the same sub-cluster AdaCos objective and cosine nearest-neighbor backend.

For all four submission slots, the frontend produces a clip embedding, the backend compares it with normal training embeddings, and the resulting distance is submitted as the anomaly score. This keeps the method path identical across systems while isolating the representation change made by each submitted variant.

4. REFERENCES

[1] DCASE Community, "DCASE 2026 Task 2: Noise-aware unsupervised anomalous sound detection for machine condition monitoring," 2026. `https://dcase.community/challenge2026/`

[2] 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," *arXiv e-prints: 2606.01578*, 2026.

[3] 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 *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2021, pp. 1–5.

[4] 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 *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, 2022.

[5] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "First-Shot Anomaly Detection for Machine Condition Monitoring: A Domain Generalization Baseline," *Proceedings of 31st European Signal Processing Conference (EUSIPCO)*, pp. 191–195, 2023.

[6] T. Fujimura, K. Wilkinghoff, K. Imoto, and T. Toda, "ASDKit: A Toolkit for Comprehensive Evaluation of Anomalous Sound Detection Methods," in *Proceedings of the 10th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2025)*, Barcelona, Spain, 2025, pp. 40–44, doi: 10.5281/zenodo.17251589.

[7] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, X. Yu, and F. Wei, "BEATs: Audio Pre-Training with Acoustic Tokenizers," in *Proceedings of ICML*, 2023.

[8] K. Wilkinghoff, "Combining Multiple Distributions Based on Sub-Cluster AdaCos for Anomalous Sound Detection Under Domain Shifted Conditions," in *Proceedings of DCASE Workshop*, 2021.

[9] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in *Proceedings of ICLR*, 2018.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.

[12] 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 *Proceedings of ICLR*, 2022.

[13] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. Cambridge, MA, USA: MIT Press, 1949.