GENREP FOR FIRST-SHOT UNSUPERVISED ANOMALOUS SOUND DETECTION OF DCASE 2025 CHALLENGE

Technical Report

Phurich Saengthong, Takahiro Shinozaki

Institute of Science Tokyo
www.ts.ip.titech.ac.jp

ABSTRACT

Recent advances in large-scale pre-trained audio models have shown that frozen embeddings can provide robust and transferable representations for general audio tasks. Building on GenRep, which uses frozen embeddings with k-nearest neighbors and domain-wise Z-score normalization for anomaly detection under domain shift, we extend this approach by exploring several directions, including normalization strategies, model scaling, and feature ensembling. First, we study alternative normalization methods such as global Z-score normalization, local density normalization, and domain-wise local density normalization. Second, we evaluate pre-trained audio encoders ranging from 5M to 300M parameters on the DCASE2025 Task 2 dataset to examine the impact of model scale. Third, we study the effect of ensemble fusion using features from multiple frozen encoders. Our results indicate that even the smallest pre-trained encoder (5.49M) can outperform a baseline autoencoder, and that larger models and ensembling contribute to further improvements without updating model parameters. The code is available open-source [1].

Index Terms— anomaly detection, acoustic condition monitoring, domain shift, first-shot problem, DCASE challenge

1. INTRODUCTION

DCASE2025 Task 2 challenge continues to focus on the first-shot problem under a domain generalization setting for anomalous sound detection (ASD), where participants must develop systems that generalize to entirely unseen machine types without tuning on target-domain data. To reflect this practical constraint, the evaluation dataset features machine types not included in the development set. Additionally, two optional resources are introduced to support performance improvement: (1) supplementary data such as clean machine sounds or noise-only recordings, and (2) external datasets from previous DCASE Task 2 challenges, to simulate the use of historical data for model pretraining in real-world scenarios [1, 2, 3, 4].

In a typical anomaly detection setting, a model is trained on normal data from an existing domain where the definition of normality is well-established. However, during deployment, the environment may change. For example, variations in background noise, operating conditions, or sensor configurations can cause a domain shift. A model trained solely on the original domain may fail to generalize under such shifts, leading to false alarms or missed anomalies. To handle domain shift, it is also necessary to define a target domain dataset that represents the new conditions [1, 5].

[1] `https://github.com/Phuriches/GenRepASD`

Recent state-of-the-art approaches for domain-generalized ASD tasks address this issue by fine-tuning or updating model parameters using the Outlier Exposure framework [6]. This typically involves training an audio encoder from scratch or fine-tuning a large-scale pre-trained audio encoder [7, 8, 9, 10, 11, 12], as illustrated in Figure 1a).

A key factor in the success of an anomaly detection system is the quality of the audio embeddings it relies on. Embeddings that robustly represent normal sound characteristics across domains are essential for distinguishing anomalies under varying conditions. It would therefore be highly beneficial to use a generic audio encoder that can extract meaningful embeddings without the need for fine-tuning, while also allowing a clear definition of the target domain. This would eliminate the need for retraining or updating the encoder during deployment, which can introduce downtime and operational complexity.

As illustrated in Figure 1b), such a system can be realized by storing training data in domain-specific memory banks that define normal behavior for both source and target domains. During inference, test data are compared against both memory banks, and the minimum distance score is selected as the final anomaly score. GenRep [13] demonstrated that embeddings extracted from large-scale pre-trained audio encoders can be effectively used for domain-generalized ASD without the need for fine-tuning. However, GenRep relies on test-time statistics for standardizing or normalizing anomaly scores, which may not always be available during inference. To address this, we explore normalization strategies that do not depend on test-time statistics, aiming to improve generalization under domain shift. Specifically, we investigate domain-wise Z-score normalization, local density normalization [12], and domain-wise local density normalization, applied to anomaly scores computed via nearest neighbor search using frozen representations from various large-scale pre-trained audio encoders. Our contributions include:

- We investigate normalization strategies—domain-wise Z-score, local density, and domain-wise local density normalization—and their effect on domain alignment and anomaly detection performance.
- We evaluate GenRep using a range of pre-trained audio encoders (5M–300M parameters) on the DCASE2025 Task 2 dataset, observing consistent gains with larger models.
- All of our systems outperform the baseline, with the best achieving an official score of 64.53. Using the smallest encoder (`ced_tiny`) with domain-wise local density normalization, remains competitive with a score of 62.15.

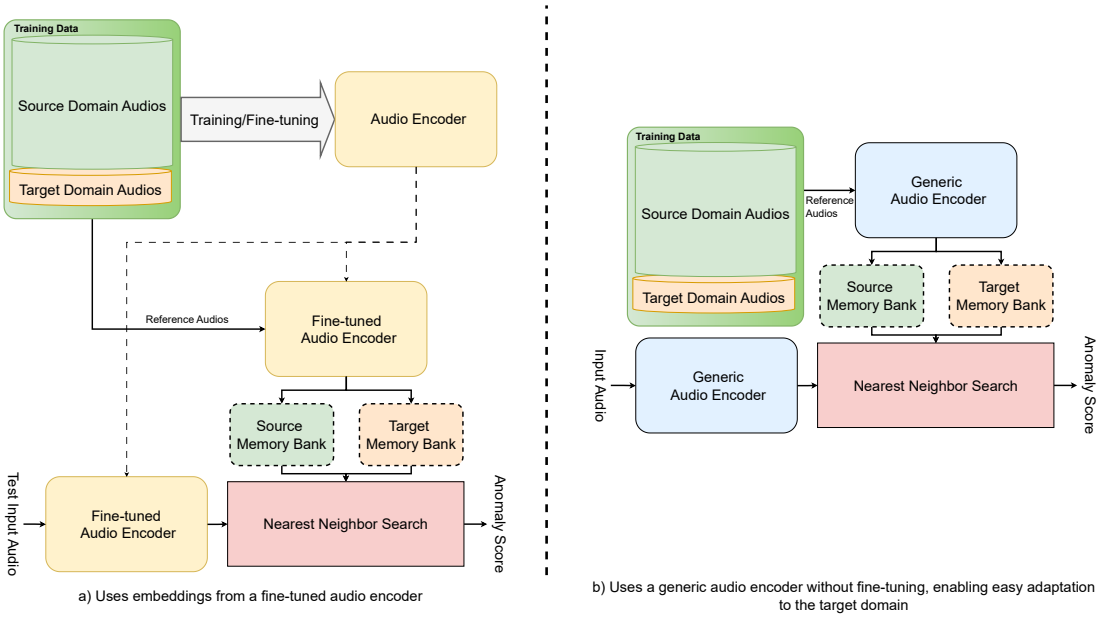

Figure 1: Comparison of anomaly detection pipelines. (a) State-of-the-art systems typically employ embeddings from a fine-tuned audio encoder. (b) A generic audio encoder is used without fine-tuning, enabling easier adaptation to the target domain.

2. APPROACH

GenRep [13] demonstrated that using generic frozen features extracted from large-scale pre-trained audio encoders can significantly improve domain-generalized anomalous sound detection (ASD). Remarkably, this approach outperforms methods that rely on fine-tuning with target-domain data [14]. A key factor contributing to this success is the use of normalization techniques. Specifically, GenRep applies Z-score standardization based on the distribution of test data.

However, this reliance on the test distribution poses a challenge in real-world deployments, where access to test data is not feasible in advance. Moreover, the DCASE2025 Task 2 challenge [1] explicitly prohibits using test data for any form of training, further highlighting the limitations of such normalization strategies.

To address this, we build upon GenRep and explore alternative normalization approaches that do not depend on the test distribution. Specifically, we focus on normalization methods that operate purely on the training data, making them better suited for domain-generalized ASD.

2.1. Domain-generalized kNN using Generic Representation

We apply GenRep [13] without using MemMixup. We store feature embeddings from the training source and target domains in memory banks, using a large-scale pre-trained audio encoder. At test time, the anomaly score of a sample y is computed by comparing its feature f_y to both memory banks.

For each domain, we calculate the average distance to the K_n nearest neighbors:

$$d(y) = \frac{1}{K_n} \sum_{f \in N_{K_n}(f_y)} \|f - f_y\|^2, \tag{1}$$

where $N_{K_n}(f_y)$ denotes the nearest neighbors in the corresponding memory bank, yielding scores $d_s(y)$ and $d_t(y)$.

Since the domain of y is unknown at inference time, we assume it belongs to the domain in which it appears most normal. To compare the scores, we apply Z-score normalization using the test-time means $\mu_s^{\text{test}}, \mu_t^{\text{test}}$ and standard deviations $\sigma_s^{\text{test}}, \sigma_t^{\text{test}}$ computed from the test anomaly score distributions. The normalized scores are defined as $\text{Z-score}(d_s) = \frac{d_s(y) - \mu_s^{\text{test}}}{\sigma_s^{\text{test}}}$ and $\text{Z-score}(d_t) = \frac{d_t(y) - \mu_t^{\text{test}}}{\sigma_t^{\text{test}}}$.

The final anomaly score is:

$$\text{score}(y) = \min\left(\frac{d_s(y) - \mu_s^{\text{test}}}{\sigma_s^{\text{test}}}, \frac{d_t(y) - \mu_t^{\text{test}}}{\sigma_t^{\text{test}}}\right). \tag{2}$$

2.2. Domain-wise Z-score normalization

To eliminate the need for test-time distribution introduced in the previous formulation, we instead estimate normalization statistics from the training data. For each training sample f_i from domain $d \in \{s, t\}$, we compute its intra-domain kNN distance as $d(f_i) = \frac{1}{K_n} \sum_{f_j \in N_{K_n}(f_i)} \|f_i - f_j\|^2$, where f_j are the K_n nearest neighbors from the same domain. This yields domain-specific training statistics: $\mu_s^{\text{train}}, \sigma_s^{\text{train}}$ and $\mu_t^{\text{train}}, \sigma_t^{\text{train}}$.

At test time, we compute $d_s(y)$ and $d_t(y)$ as before, and normalize them using the training-based means. While we preserve domain-specific means, we empirically find that using the **target domain's standard deviation** σ_t^{train} for both normalizations improves performance by aligning scores to a common scale. The final anomaly score becomes:

$$\text{score}(y) = \min\left(\frac{d_s(y) - \mu_s^{\text{train}}}{\sigma_t^{\text{train}}}, \frac{d_t(y) - \mu_t^{\text{train}}}{\sigma_t^{\text{train}}}\right). \tag{3}$$

2.3. Local Density Normalization

We further investigate applying a local density normalization approach [12] that adjusts the anomaly score based on the density around each reference sample, applying to the GenRep framework. For a test sample feature f_y and a set of reference features $\mathcal{F}_{\text{ref}}$ (e.g., from a memory bank), the locally normalized anomaly score is defined as:

$$\text{score}(y) = \min_{f \in \mathcal{F}_{\text{ref}}} \frac{d(f_y, f)}{\sum_{k=1}^{K} d(f, f_k)}, \tag{4}$$

where $d(f_y, f)$ is the distance between the test sample and a reference feature, and f_k denotes the K nearest neighbors of f in $\mathcal{F}_{\text{ref}}$. This formulation rescales the anomaly score by the local density around the reference feature.

2.4. Domain-wise Local Density Normalization

We also explore a possible extension of local density normalization [12] by applying it in a domain-wise manner. Specifically, for each test sample feature f_y, we compute its locally normalized anomaly scores with respect to both the source memory bank $\mathcal{F}_s$ and the target memory bank $\mathcal{F}_t$. Each score is adjusted based on the local density around reference features within the corresponding domain. To accommodate imbalanced reference sizes, such as when the target domain contains significantly fewer samples, we allow the number of neighbors K to differ by domain, denoted as K_s for the source and K_t for the target. The final anomaly score is then obtained by taking the minimum of the two normalized scores:

$$\text{score}(y) = \min \left(\min_{f \in \mathcal{F}_s} \frac{d(f_y, f)}{\sum_{k=1}^{K_s} d(f, f_k)}, \min_{f \in \mathcal{F}_t} \frac{d(f_y, f)}{\sum_{k=1}^{K_t} d(f, f_k)} \right), \tag{5}$$

where f_k denotes the K_s or K_t nearest neighbors of f within its respective domain-specific memory bank.

2.5. Anomaly Detection Details

We investigate five state-of-the-art large-scale pre-trained audio encoders within the GenRep framework, denoted as follows: `BEATs_ft1` [2] for BEATs [15], `m2d_clap` [3] for M2D CLAP [16], `EAT_large` [4] for EAT [17], `SSLAM` [5] for SSLAM [18], and `ced_base` and `ced_tiny` [6] for CED [19]. For each encoder, we extract features from the training data and store them in the corresponding source and target memory banks. No supplemental data are used in this process.

Table 1 summarizes the configurations of our submitted systems, including normalization methods, feature layers used, and model complexity. To submit our evaluation scores to the challenge, we adopted a standard ensemble strategy by averaging the anomaly scores produced by GenRep using five frozen audio encoders: `BEATs_ft1`, `m2d_clap`, `SSLAM`, `EAT_large`, and `ced_tiny`.

[2] https://github.com/microsoft/unilm/tree/master/beats

[3] https://github.com/nttcslab/m2d

[4] https://github.com/cwx-worst-one/EAT

[5] https://github.com/ta012/SSLAM/

[6] https://github.com/RicherMans/CED

System	Normalization	Feature Layers	MACs / Params
System 1	Domain-wise LD	Last two layers	271.71 G / 569.28 M
System 2	LD	Last two layers	271.71 G / 569.28 M
System 3	Domain-wise Z-score	Layer 7 and 10*	271.71 G / 569.28 M
System 4	Domain-wise LD	Layer 7 and 10	1.34 G / 5.49 M

Table 1: System's GenRep configurations including normalization method, feature layers used, and model complexity. LD = Local density. *For `EAT_large`, System 3 uses the last two layers.

We applied three different score normalization methods to form three ensemble systems: **System 1** uses domain-wise local density normalization, **System 2** applies local density normalization without domain separation, and **System 3** adopts domain-wise Z-score normalization. Each ensemble system shares the same model complexity of 271.71 G MACs and 569.28 M parameters. Additionally, we submitted a lightweight variant, **System 4**, which uses only `ced_tiny` and domain-wise local density normalization, resulting in a compact configuration with 1.34 G MACs and 5.49 M parameters.

For normalization parameters, we set $K = 1$ for domain-wise Z-score normalization, using $K = 1$ for both source and target domains. We set $K = 16$ for local density normalization, and $K_{\text{s}} = 16$, $K_{\text{t}} = 9$ for domain-wise local density normalization.

Dataset summary. The dataset comprises three subsets: development, additional training, and evaluation datasets. The development dataset includes seven machine types, each with one section containing 990 normal clips from a source domain, 10 normal clips from a target domain, and 200 labeled test clips (100 normal and 100 anomalous) with domain labels. Some machines also include attribute annotations. The additional training dataset introduces nine new machine types, each with the same training structure, though attributes are provided for only some machines. The evaluation dataset includes test clips corresponding to the additional training machines, without any labels or domain information. Participants are required to train models using only one section per machine type, without tuning on the test set or relying on attribute information [1].

Evaluation metrics. Performance under domain shift is evaluated using AUC, partial AUC (pAUC), and the Official Score, which is defined as the harmonic mean of source AUC, target AUC, and mixed pAUC across all machine types [1].

3. EXPERIMENTAL RESULTS

System	AUC Source	AUC Target	pAUC	Official Score
Baseline [4]	66.78	51.39	52.94	56.26
System1	**76.11**	61.66	58.36	**64.53**
System2	63.42	**68.73**	**59.69**	63.74
System3	67.96	61.46	56.23	61.51
System4	72.56	60.01	56.10	62.15

Table 2: Comparison of baseline and submitted systems on development data. Best results per column are highlighted in bold.

As shown in Table 2, **System1**, which applies domain-wise local density normalization, achieves the highest overall performance with an official score of 64.53 and the best AUC Source (76.11). **System2**, which uses local density normalization without domain

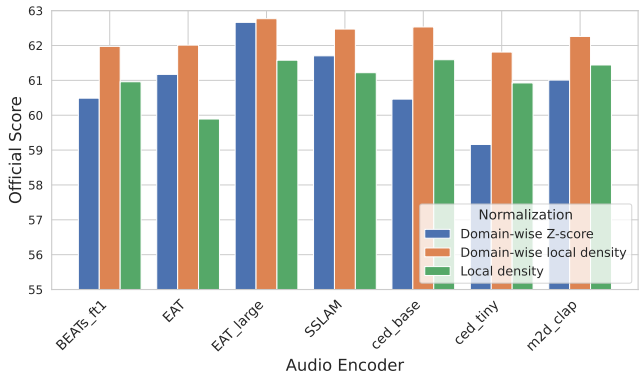

Figure 2: Comparison of normalization scoring approaches. For each audio encoder, we report the performance using the layer that achieves the best overall performance across all machine types (rather than cherry-picking the best-performing layer for each individual machine, which would result in inflated performance).

separation, ranks second with an official score of 63.74, while also achieving the best results for AUC Target (68.73) and pAUC (59.69). **System4**, a lightweight variant using only `ced_tiny` with domain-wise local density normalization, achieves a strong official score of 62.15, slightly outperforming **System3**, which applies domain-wise Z-score normalization and scores 61.51. The baseline system [4] performs the lowest with an official score of 56.26, well below all proposed systems.

Figure 2 presents the official scores for three normalization methods—domain-wise Z-score (blue), domain-wise local density normalization (orange), and local density normalization (green)—across seven audio encoders. Domain-wise local density normalization (orange) generally achieves the highest or comparable scores for most encoders. Domain-wise Z-score normalization (blue) maintains relatively consistent performance, while local density normalization without domain separation (green) shows greater variability and lower scores in some cases, such as `EAT` and `ced_tiny`. The performance gap between normalization methods differs by encoder: some encoders, like `EAT_large`, `SSLAM`, and `ced_base`, exhibit minor differences, whereas others, such as `EAT` and `ced_tiny`, show more pronounced variations.

4. DISCUSSION AND CONCLUSION

We believe our approach provides a practical and effective baseline for future work in domain-generalized anomalous sound detection. Despite its simplicity, using frozen audio encoders and lightweight normalization techniques, our system achieves strong performance across diverse machine types without relying on test-time adaptation. Notably, even compact models such as `CED_tiny`, when combined with domain-wise local density normalization, surpass the performance of the conventional autoencoder-based baseline [4], highlighting the potential of off-the-shelf representations in challenging domain shift scenarios.

Machine	System 1	System 2	System 3	System 4
ToyCar				
AUC source	70.90	63.12	64.60	70.54
AUC target	67.72	71.56	71.10	63.44
pAUC	54.37	55.68	52.21	50.53
Official	63.47	62.79	61.60	60.32
ToyTrain				
AUC source	88.34	84.34	79.14	87.12
AUC target	68.06	69.88	69.62	67.52
pAUC	60.53	59.79	54.53	55.21
Official	70.53	69.94	66.17	67.57
bearing				
AUC source	71.12	70.12	65.12	64.62
AUC target	60.28	61.86	57.94	53.98
pAUC	60.21	60.32	60.84	56.74
Official	63.48	63.82	61.16	58.11
fan				
AUC source	72.44	34.26	58.88	67.88
AUC target	47.28	74.72	47.26	48.98
pAUC	51.63	56.95	50.95	49.00
Official	55.23	49.89	51.93	53.00
gearbox				
AUC source	70.30	66.92	63.66	67.04
AUC target	59.38	69.70	56.56	58.42
pAUC	56.21	58.74	52.84	57.63
Official	61.41	64.77	57.35	60.75
slider				
AUC source	82.00	82.54	77.78	78.78
AUC target	57.60	57.92	57.86	59.30
pAUC	56.58	57.05	55.47	59.89
Official	63.52	63.95	62.28	64.86
valve				
AUC source	81.52	82.24	71.50	76.96
AUC target	82.46	80.68	82.72	76.00
pAUC	73.58	72.00	71.47	67.58
Official	78.98	78.04	74.88	73.26
All (Avg)				
AUC source	76.11	63.42	67.96	72.56
AUC target	61.66	68.73	61.46	60.01
pAUC	58.36	59.69	56.23	56.10
Official	64.53	63.74	61.51	62.15

Table 3: Anomaly detection performance across systems and machines on the development set. Official score is the harmonic mean of AUC source, AUC target, and pAUC.

Table 4: Model complexity comparison.

Model	MACs (G)	Params (M)
Baseline	0.17	0.27
BEATs_ft1	45.01	90.71
m2d_clap	26.50	85.25
EAT	43.71	85.25
SSLAM	43.71	85.25
ced_tiny	1.33	5.49
ced_base	21.13	85.66
EAT_large	155.16	302.57

5. REFERENCES

[1] T. Nishida, N. Harada, D. Niizumi, D. Albertini, R. Sannino, S. Pradolini, F. Augusti, K. Imoto, K. Dohi, H. Purohit, T. Endo, and Y. Kawaguchi, "Description and discussion on DCASE 2025 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," *In arXiv e-prints: 2506.10097*, 2025.

[2] 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 *DCASE*, Barcelona, Spain, November 2021.

[3] 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)*, Nancy, France, November 2022.

[4] 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.

[5] K. Wilkinghoff, T. Fujimura, K. Imoto, J. L. Roux, Z.-H. Tan, and T. Toda, "Handling domain shifts for anomalous sound detection: A review of dcase-related work," 2025. [Online]. Available: https://arxiv.org/abs/2503.10435

[6] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," *ICLR*, 2019.

[7] K. Wilkinghoff, "Self-supervised learning for anomalous sound detection," in *ICASSP*. IEEE, 2024, pp. 276–280.

[8] Z. Lv, A. Jiang, B. Han, Y. Liang, Y. Qian, X. Chen, J. Liu, and P. Fan, "Aithu system for first-shot unsupervised anomalous sound detection," DCASE2024 Challenge, Tech. Rep., June 2024.

[9] 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.

[10] A. Jiang, X. Zheng, B. Han, Y. Qiu, P. Fan, W.-Q. Zhang, C. Lu, and J. Liu, "Adaptive prototype learning for anomalous sound detection with partially known attributes," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[11] T. Fujimura, I. Kuroyanagi, and T. Toda, "Improvements of discriminative feature space training for anomalous sound detection in unlabeled conditions," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[12] K. Wilkinghoff, H. Yang, J. Ebbers, F. G. Germain, G. Wichern, and J. L. Roux, "Keeping the balance: Anomaly score calculation for domain generalization," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[13] P. Saengthong and T. Shinozaki, "Deep generic representations for domain-generalized anomalous sound detection," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[14] 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," May 2023, arXiv:2305.07828 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2305.07828

[15] 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 *ICML*, vol. 202. PMLR, July 2023. [Online]. Available: https://proceedings.mlr.press/v202/chen23ag.html

[16] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, M. Yasuda, S. Tsubaki, and K. Imoto, "M2D-CLAP: Masked Modeling Duo Meets CLAP for Learning General-purpose Audio-Language Representation," *to appear at Interspeech*, 2024. [Online]. Available: https://arxiv.org/abs/2406.02032

[17] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "Eat: Self-supervised pre-training with efficient audio transformer," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, K. Larson, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2024, pp. 3807–3815, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2024/421

[18] T. Alex, S. Atito, A. Mustafa, M. Awais, and P. J. B. Jackson, "SSLAM: Enhancing self-supervised models with audio mixtures for polyphonic soundscapes," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=odU59TxdiB

[19] H. Dinkel, Y. Wang, Z. Yan, J. Zhang, and Y. Wang, "Ced: Consistent ensemble distillation for audio tagging," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.