MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Mind the Gap: Detecting Cluster Exits for Robust Local Density-Based Score Normalization in Anomalous Sound Detection

Wilkinghoff, Kevin; Wichern, Gordon; Le Roux, Jonathan; Tan, Zheng-Hua

TR2026-138 September 29, 2026

Abstract

Local density-based score normalization is an effective component of distance-based embedding methods for anomalous sound detection, particularly when data densities vary across conditions or domains. In practice, however, performance depends strongly on neighborhood size. Increasing it can degrade detection accuracy when neighborhood expansion crosses cluster boundaries, violating the locality assumption of local density estimation. This observation motivates adapting the neighborhood size based on locality preservation rather than fixing it in advance. We realize this by proposing cluster exit detection, a lightweight mechanism that identifies distance discontinuities and selects neighborhood sizes accordingly. Experiments across multiple embedding models and datasets show improved robustness to neighborhood-size selection and consistent performance gains.

Interspeech 2026

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Mind the Gap: Detecting Cluster Exits for Robust Local Density-Based Score Normalization in Anomalous Sound Detection

Kevin Wilkinghoff [1,2,**], *Gordon Wichern* [3], *Jonathan Le Roux* [3], *Zheng-Hua Tan* [1,2]

[1] Department of Electronic Systems, Aalborg University, Aalborg, Denmark
[2] Pioneer Centre for Artificial Intelligence, Copenhagen, Denmark
[3] Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

kevin.wilkinghoff@ieee.org, wichern@merl.com, leroux@merl.com, zt@es.aau.dk

Abstract

Local density-based score normalization is an effective component of distance-based embedding methods for anomalous sound detection, particularly when data densities vary across conditions or domains. In practice, however, performance depends strongly on neighborhood size. Increasing it can degrade detection accuracy when neighborhood expansion crosses cluster boundaries, violating the locality assumption of local density estimation. This observation motivates adapting the neighborhood size based on locality preservation rather than fixing it in advance. We realize this by proposing cluster exit detection, a lightweight mechanism that identifies distance discontinuities and selects neighborhood sizes accordingly. Experiments across multiple embedding models and datasets show improved robustness to neighborhood-size selection and consistent performance gains.

Index Terms: anomaly detection, anomalous sound detection, domain generalization, domain shift, score normalization

1. Introduction

Semi-supervised anomalous sound detection (ASD) for machine condition monitoring aims to detect anomalous machine sounds using training data that contain only recordings of normal operation. In real-world deployment, however, systems often encounter domain shifts, such as changes in operating conditions, background noise, recording devices, or environments, which alter the sound distribution even though the machine remains in a normal state. This setting is formalized in recent DCASE challenges [1–4], where models are trained with many normal samples from a source domain and only a few from a target domain, yet are expected to perform equally well in both domains at test time.

To handle this setting, many state-of-the-art ASD systems operate in learned embedding spaces using distance-based scores or density estimates. Each audio recording, typically several seconds long, is mapped to a fixed-dimensional embedding vector by a neural network, either trained with task-specific surrogate objectives for machine sounds [5–10] or pre-trained on large-scale audio datasets [11–17]. Anomalies are identified by comparing embeddings of test samples to those of normal training data in the embedding space.

Recently, local density-based anomaly score normalization (LDN) [18, 19] has been proposed to improve robustness to domain shifts in this embedding-based setting [20]. LDN normalizes distance-based anomaly scores using statistics from a local neighborhood in the embedding space, compensating for variations in local data density. This is particularly important when detection relies on a single, domain-agnostic threshold despite highly non-uniform reference densities. However, LDN introduces the neighborhood size as a critical hyperparameter.

Empirically, increasing the neighborhood size beyond one or two neighbors often leads to systematic performance degradation. While larger neighborhoods are generally expected to yield more stable density estimates, existing approaches typically fix the neighborhood size to a small value [19, 21], offering limited insight into why performance deteriorates as the neighborhood expands. Variance minimization (VarMin) [22] can partially mitigate this effect, but the fundamental sensitivity to neighborhood size remains unresolved.

In this work, we show that the observed degradation is not caused by larger neighborhoods per se, but occurs when neighborhood expansion crosses cluster boundaries in the embedding space. Once neighbors outside the local region are included, the locality assumption underlying LDN is violated, corrupting local density estimates and destabilizing score normalization. This insight motivates a general design principle: Neighborhood sizes should adapt based on whether locality is preserved, rather than being fixed a priori. Building on this principle, we propose cluster-exit detection (CED), a lightweight, training-free mechanism that detects distance jumps indicating cluster exits and adapts neighborhood sizes on a per-sample basis.

The main contributions of this work are:

- We identify a structural failure mode of LDN arising from neighborhood expansion across cluster boundaries and motivate the need for adaptive neighborhood selection to preserve local density structure.
- We propose CED, a lightweight, training-free algorithm for adaptive neighborhood selection based on distance jumps.
- We conduct experiments using five embedding models across five benchmark datasets, demonstrating improved robustness to neighborhood-size selection and consistent performance gains over fixed, small-neighborhood baselines.

2. Local density-based anomaly score normalization

LDN has been shown to improve robustness to domain shifts in distance-based anomaly detection [18, 19]. The key idea is to normalize anomaly scores using statistics derived from a local neighborhood in the embedding space, thereby accounting for variations in local data density. More recently, VarMin was proposed to further stabilize normalized scores [22].

We briefly recap LDN, its variance-minimized variant, and the role of the neighborhood size.

** indicates the corresponding author.

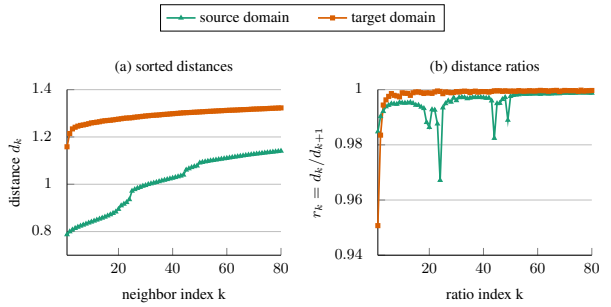

Figure 1: *Average sorted distances (left) and distance ratios (right) for BEATs embeddings of the "ToyCar" machine on the DCASE2025 dataset in the source and target domains. Pronounced distance jumps and low ratios mark cluster exits, which occur earlier in the target domain and reveal violations of locality under fixed neighborhood sizes.*

2.1. Base anomaly scoring

Let $\mathcal{X}_{\text{test}} \subset \mathbb{R}^D$ denote the set of test samples and $\mathcal{X}_{\text{ref}} \subset \mathbb{R}^D$ a reference set of normal training samples in the embedding space. All training samples from both domains are used as reference samples. For a test sample $x \in \mathcal{X}_{\text{test}}$, the base anomaly score is computed as the distance to its closest reference sample,

$$\mathcal{A}(x, \mathcal{X}_{\text{ref}}) := \min_{y \in \mathcal{X}_{\text{ref}}} \mathcal{D}(x, y) \in \mathbb{R}_+, \tag{1}$$

where $\mathcal{D} : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}_+$ denotes a distance measure such as the Euclidean distance or cosine distance.

2.2. Local density-based normalization

To mitigate performance degradation caused by domain shifts, LDN normalizes the raw anomaly score using distances within a local neighborhood of each reference sample. For a given $y \in \mathcal{X}_{\text{ref}}$, a neighborhood of size $K \in \mathbb{N}$ is defined by its K nearest neighbors in the reference set. Increasing K implicitly assumes that locality is preserved as the neighborhood expands.

Combined with VarMin in log-space, this yields scaled anomaly scores of the form

$$\mathcal{A}_{\text{scaled}}(x, \mathcal{X}_{\text{ref}} \mid K, \alpha) := \min_{y \in \mathcal{X}_{\text{ref}}} \Big(\log \mathcal{D}(x, y) - \alpha \log \frac{1}{K} \sum_{k=1}^{K} \mathcal{D}(y, y_k) \Big) \in \mathbb{R}, \tag{2}$$

where y_k denotes the k-th nearest neighbor of y in $\mathcal{X}_{\text{ref}}$. Setting $\alpha = 1$ recovers the standard LDN formulation.

For VarMin, $\alpha = \alpha^*$ is selected to minimize the variance of the normalized scores over the reference set, i.e.,

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}} \operatorname{Var}_{z \sim \mathcal{X}_{\text{ref}}} \left(\mathcal{A}_{\text{scaled}}\left(z, \mathcal{X}_{\text{ref}} \mid K, \alpha\right) \right) \in \mathbb{R}. \tag{3}$$

This scoring backend is entirely training-free, requires no labels, and introduces no additional assumptions on the data distribution. Since the normalization constants depend only on the reference samples, they can be pre-computed without additional inference-time overhead.

3. Locality violations and cluster exits

LDN assumes that nearby reference samples belong to the same local region in the embedding space. Increasing the neighborhood size K therefore assumes that this locality remains valid as the neighborhood expands. In practice, LDN is applied with very small neighborhoods, since increasing K beyond one or two neighbors often leads to systematic performance degradation, even though larger neighborhoods are expected to yield more stable density estimates. As shown in Fig. 1(a), this degradation is structural rather than statistical. Especially in the target domain, large distance jumps already occur among the first few neighbors. Sorted distance profiles exhibit pronounced jumps when neighborhood expansion crosses cluster boundaries, directly indicating a violation of the locality assumption. The resulting sensitivity to K is further quantified in Fig. 2.

These jumps explain how locality is lost. For a given reference sample, distances typically increase smoothly within the same cluster. When expansion crosses a cluster boundary, this smooth increase is interrupted by a sharp jump in distance. We refer to this transition as a *cluster exit*. As long as distances grow smoothly, neighborhood expansion remains valid. Once a cluster exit is encountered, further expansion includes samples from outside the local region, which corrupts local density estimates and degrades score normalization. The core challenge is therefore not selecting a fixed neighborhood size, but identifying when locality no longer holds. In the following section, we present a training-free algorithm that detects cluster exits from distance jumps and uses them to adapt neighborhood sizes for local density estimation. While developed for LDN, the underlying principle is more general, and alternative mechanisms for detecting cluster exits could be integrated within the same framework.

4. Cluster exit detection

CED is a training-free mechanism for adaptive neighborhood selection in local density-based normalization. Because local density estimates in LDN are defined with respect to reference neighborhoods, all computations are carried out independently for each reference sample $y \in \mathcal{X}_{\text{ref}}$ using only distances to other reference samples. The method requires no labels or training, and can replace a fixed neighborhood size K in existing LDN backends. The central contribution is to demonstrate that adaptively selecting neighborhood sizes via cluster exit detection, rather than fixing K, is essential for preserving locality and stabilizing performance. Fixed thresholds are used throughout all experiments.

4.1. Distance ratios

Let $K \geq 2$ and let $\{y_k\}_{k=1}^{K}$ denote the K nearest neighbors of a reference sample $y \in \mathcal{X}_{\text{ref}}$, ordered by increasing distance. Define

$$d_k(y) := \mathcal{D}(y, y_k), \quad k = 1, \ldots, K. \tag{4}$$

Within a local cluster, distances increase smoothly, whereas crossing a cluster boundary induces a sharp increase. We capture such changes using distance ratios

$$r_k(y) = \frac{d_k(y)}{d_{k+1}(y) + \epsilon}, \quad k = 1, \ldots, K-1, \tag{5}$$

with $\epsilon = 10^{-12}$. Since distances are ordered, $0 < r_k(y) \leq 1$, and small values indicate potential cluster exits. For $K > 2$, adjacent ratios are averaged,

$$\tilde{r}_k(y) = \tfrac{1}{2}(r_k(y) + r_{k+1}(y)), \quad k = 1, \ldots, K-2, \tag{6}$$

to reduce sensitivity to isolated fluctuations. For $K = 2$, no smoothing is applied. The corresponding distance ratios are

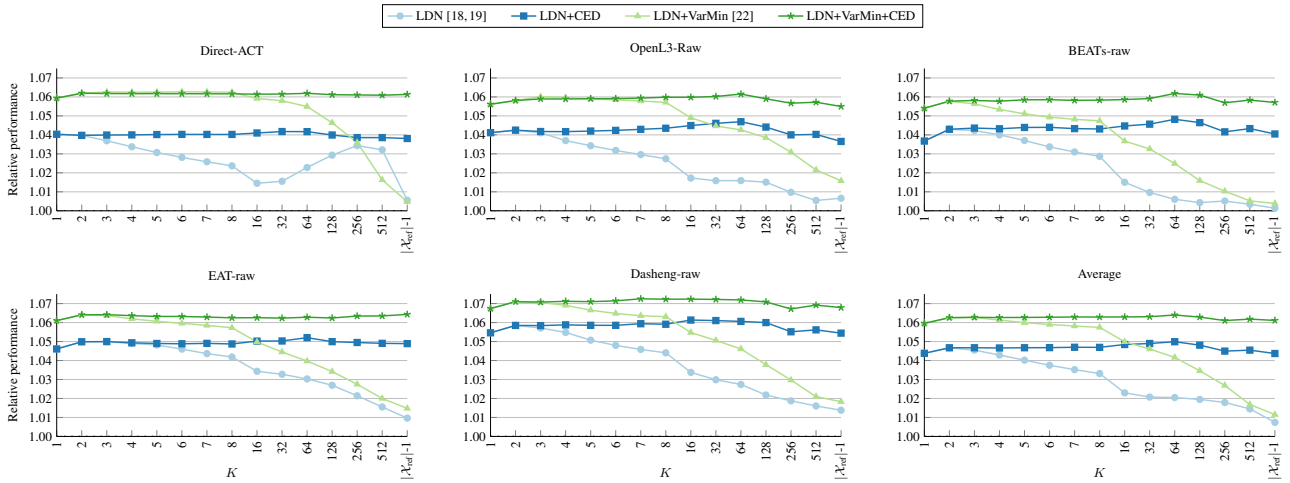

Figure 2: *Relative performance with respect to not applying LDN, shown as a function of the neighborhood size K. Values correspond to the ratio between the performance obtained with the respective normalization method and that obtained without LDN. Results are geometric means across all five evaluated datasets. For Direct-ACT, results are averaged over ten independent trials.*

shown in Fig. 1(b). Pronounced drops in the ratio sequence indicate potential cluster exits.

4.2. Detecting cluster exits

As the neighborhood size k increases, a cluster exit is reflected by a pronounced drop in the smoothed ratio sequence $\tilde{r}_k(y)$. We therefore first identify the most informative transition by locating the index of the smallest ratio value,

$$k_{\min}(y) = \underset{k\in\{1,\dots,K-2\}}{\arg\min} \; \tilde{r}_k(y). \tag{7}$$

In practice, neighborhood expansion may already become unreliable at earlier indices if $\tilde{r}_k(y)$ drops below a conservative, data-adaptive threshold. To account for this, we define

$$\mathcal{C}(y) = \Big\{k \in \{1,\dots,K-2\} : \quad \tilde{r}_k(y) < Q_{0.04}(\tilde{\mathbf{r}}(y))\Big\} \tag{8}$$

where $Q_{0.04}(\tilde{\mathbf{r}}(y))$ denotes the 4th percentile of the ratio sequence $\tilde{\mathbf{r}}(y)$. If $\mathcal{C}(y)$ is nonempty, neighborhood expansion is truncated at the earliest candidate,

$$k_{\text{ext}}(y) = \min \mathcal{C}(y), \tag{9}$$

otherwise we set $k_{\text{ext}}(y) = K-2$. The adaptive neighborhood size is then chosen as

$$\hat{K}(y) = \min\{k_{\text{ext}}(y),\, k_{\min}(y)\} + 1. \tag{10}$$

Since ratios require neighbor pairs, the index is effectively shifted, increasing the adaptive neighborhood size by one.

4.3. Conservative fallback for sparse regions

To avoid unreliable truncation in weakly structured regions, we first summarize the overall behavior of the ratio sequence by $r_{\min}(y) = \min_k r_k(y)$. Very strong ratio drops indicate that the reference point y is located in a highly sparse region. In such cases, we cautiously fall back to two neighbors when

$$r_1(y) < 0.85 \quad \text{or} \quad \frac{r_1(y)}{r_{\min}(y)} > 1.02. \tag{11}$$

We use two neighbors instead of one for consistency with the baseline LDN without CED.

4.4. Integration into local density normalization

Given the adaptive neighborhood size $\hat{K}(y)$, the local density estimate is computed as

$$\mu(y) = \frac{1}{\hat{K}(y)} \sum_{k=1}^{\hat{K}(y)} \mathcal{D}(y, y_k), \tag{12}$$

and replacing the fixed neighborhood size in LDN yields the final scaled score

$$\mathcal{A}^{\text{CED}}_{\text{scaled}}(x, \mathcal{X}_{\text{ref}} \mid \alpha) = \min_{y\in\mathcal{X}_{\text{ref}}} \big(\log \mathcal{D}(x, y) - \alpha \log \mu(y)\big). \tag{13}$$

5. Experimental setup

5.1. Datasets

We evaluate performance on five publicly available datasets for semi-supervised acoustic anomaly detection. We consider the DCASE2020 dataset [23], constructed from MIMII [24] and ToyADMOS [25]; DCASE2022 [1], based on MIMII-DG [26] and ToyADMOS2 [27]; DCASE2023 [2], extending MIMII-DG and ToyADMOS2+ [28]; DCASE2024 [3], incorporating MIMII-DG, ToyADMOS2# [29], and recordings collected under the IMAD-DS setup [30]; and DCASE2025 [4], consisting of MIMII-DG, ToyADMOS2025 [31], and additional IMAD-DS recordings. All datasets address semi-supervised ASD for machine condition monitoring across multiple machine types. They are partitioned into development and evaluation sets, with training splits containing only normal samples and test splits including both normal and anomalous recordings. Except for DCASE2020, which has a single domain, the datasets are designed to assess domain generalization by providing 990 and 10 source and target training samples per machine, respectively. In the test sets, machine types are known and domains are balanced, but explicit domain labels are omitted.

All experiments follow the official evaluation protocols. For DCASE2020, we report the arithmetic mean of the AUC and pAUC [32] with $p = 0.1$. For the remaining datasets, we report the harmonic mean of domain-specific AUCs and the domain-independent pAUC.

Table 1: *Average performance for different normalization approaches across the development and evaluation sets of the DCASE2020, DCASE2022, DCASE2023, DCASE2024, and DCASE2025 ASD datasets. Δ denotes improvement over baseline. CIs are 95% paired-bootstrap intervals. Highest numbers in each column are in bold.*

		Embedding Model											
		Direct-ACT		OpenL3		BEATs		EAT		Dasheng		**Average Performance**	
Normalization	**K**	average	Δ(CI)	average	Δ(CI)	average	Δ(CI)	average	Δ(CI)	average	Δ(CI)	average	Δ(CI)
-	-	65.58%	-	61.77%	-	65.02%	-	61.82%	-	60.27%	-	62.90%	-
LDN	2	67.79%	baseline	64.62%	baseline	68.04%	baseline	64.99%	baseline	64.07%	baseline	65.90%	baseline
LDN+CED	64	67.91%	+0.12 [0.02, 0.24]	64.88%	+0.26 [0.05, 0.47]	68.36%	+0.32 [0.03, 0.64]	65.14%	+0.12 [-0.10, 0.34]	64.18%	+0.11 [-0.04, 0.27]	66.09%	+0.19 [0.03, 0.35]
LDN+VarMin	2	**69.25%**	baseline	65.56%	baseline	68.95%	baseline	**65.88%**	baseline	64.81%	baseline	66.89%	baseline
LDN+VarMin+CED	64	69.24%	-0.01 [-0.07, 0.08]	**65.75%**	+0.19 [-0.01, 0.42]	**69.20%**	+0.25 [-0.01, 0.53]	65.8%	-0.08 [-0.17, 0.00]	**64.86%**	+0.05 [-0.10, 0.19]	**66.96%**	+0.07 [-0.04, 0.20]

Table 2: *Per-dataset improvements of CED over the respective baseline (cf. Table 1). Δ_{avg} denotes the mean improvement across embeddings (each averaged over splits), Δ_{max} the largest single-split gain for any embedding, and #Emb↑ the number of embeddings (out of 5) with positive average performance gain.*

	without VarMin			**with VarMin**		
Dataset	Δ_{avg}	Δ_{max}	#Emb↑	Δ_{avg}	Δ_{max}	#Emb↑
DCASE2020	-0.06%	$+0.13\%$	2/5	-0.12%	$+0.01\%$	1/5
DCASE2022	$+0.29\%$	$+0.55\%$	5/5	$+0.10\%$	$+0.37\%$	4/5
DCASE2023	$+0.32\%$	$+\mathbf{1.39}\%$	5/5	$+0.19\%$	$+\mathbf{1.12}\%$	4/5
DCASE2024	$+0.18\%$	$+0.68\%$	4/5	$+0.13\%$	$+0.64\%$	4/5
DCASE2025	$+0.20\%$	$+0.82\%$	4/5	$+0.06\%$	$+0.72\%$	4/5

5.2. Embedding models and baselines

We evaluate five embedding models spanning task-specific and large-scale pre-trained representations. We include the Direct-ACT model [19], based on [33] and trained with the AdaProj loss [34] using a subspace dimension of 32. It employs FFT- and STFT-based feature branches and is trained with an auxiliary classification task (ACT) objective over combined machine ID and attribute classes, complemented by a self-supervised feature exchange task. We further include 512-dimensional openL3 embeddings [35] pre-trained on environmental sounds, BEATs [36] trained for three iterations on AudioSet [37], EAT [38] pre-trained for 20 epochs on AudioSet, and the base Dasheng model [39]. Following [17], embeddings are aggregated using weighted generalized mean pooling with $p = 3$ and relative deviation pooling weights ($\gamma = 8$ for openL3, $\gamma = 16$ for BEATs, $\gamma = 1$ for EAT, and $\gamma = 20$ for Dasheng). For EAT, embeddings are additionally pre-processed by thresholding low-valued components at 0.1 and suppressing activation spikes via soft clipping using $x \mapsto \tanh(x/0.5) \cdot 0.5$.

As baselines, we consider plain LDN [18, 19] and LDN combined with VarMin [22], following Section 2. Cosine distance is used for Direct-ACT, while mean squared error is employed for all pre-trained embeddings.

6. Results and discussion

6.1. Sensitivity to neighborhood size

We first analyze the sensitivity of existing approaches, namely LDN and LDN+VarMin, to the choice of the neighborhood size. As shown in Fig. 2, average performance improves only when increasing the neighborhood size K from 1 to 2. This observation is consistent with the findings in [19], which recommended using $K = 1$ as a conservative default value for estimating the local neighborhood. When increasing K further, performance generally decreases monotonically and eventually approaches the level of not applying LDN, except for Direct-ACT without VarMin. This behavior is observed for both LDN with and without VarMin and verifies the claims made in Section 3.

6.2. Effect of cluster exit detection

Figure 2 further illustrates the effect of cluster exit detection on the sensitivity of LDN to the neighborhood size K. Across all embedding models, CED markedly stabilizes performance for both plain LDN and LDN+VarMin. Whereas baseline performance typically peaks at $K \in 1, 2$ and degrades for larger neighborhoods, CED enables robust performance across a wide range of K, with optimal values shifting to $K \in 16, 32, 64$. This effect is more pronounced without VarMin. Since VarMin already compensates for part of the variability introduced by imperfect local-density estimates, the additional gains obtained from CED are naturally smaller when both methods are combined. Large fixed neighborhoods alone do not yield similar gains, indicating that improvements stem from adaptive truncation rather than from increasing K itself.

Quantitative gains are summarized in Table 1 and broken down per dataset in Table 2. Although global averages are moderate, improvements exhibit a clear structure across datasets. No systematic gains are observed on DCASE2020, consistent with its homogeneous data distribution and absence of explicit domain shifts, which reduce opportunities for pronounced local sub-cluster structure. In contrast, on DCASE2022–2025, CED improves performance for at least 4 out of 5 embedding models, with average gains up to $+0.32\%$ and peak single-split improvements reaching $+1.39\%$ (DCASE2023). Without VarMin, improvements are frequently statistically significant. With VarMin, gains are smaller but remain structured on DCASE2022–2025, with peaks up to $+1.12\%$ (DCASE2023). Overall, these results indicate that adaptive locality preservation is particularly beneficial under heterogeneous local structure.

7. Conclusion

In this work, we analyzed the sensitivity of LDN to neighborhood size and identified a failure mode that occurs when neighborhood expansion crosses cluster boundaries, thereby violating the locality assumption underlying density estimation. Based on this insight, we proposed CED, a lightweight mechanism that detects distance discontinuities to identify neighborhood exits and adapt score normalization accordingly. Experiments across multiple embedding models and benchmark datasets showed that CED reduces sensitivity to neighborhood-size selection and improves performance over a wide range of neighborhood sizes. These results indicate that the observed performance degradation is not an inherent limitation of LDN, but a consequence of silently violated locality assumptions.

8. Generative AI disclosure

Generative AI tools were used for language editing and polishing of the manuscript. All scientific content, interpretations, and conclusions are the responsibility of the authors.

9. References

[1] K. Dohi *et al.*, "Description and discussion on DCASE 2022 Challenge Task 2: Unsupervised anomalous sound detection for machine condition monitoring applying domain generalization techniques," in *Proc. DCASE*, 2022.

[2] ——, "Description and discussion on DCASE 2023 Challenge Task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2023.

[3] T. Nishida *et al.*, "Description and discussion on DCASE 2024 Challenge Task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2024.

[4] ——, "Description and discussion on DCASE 2025 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2025.

[5] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Proc. DCASE*, 2020.

[6] P. Primus, V. Haunschmid, P. Praher, and G. Widmer, "Anomalous sound detection as a simple binary classification problem with careful selection of proxy outlier examples," in *Proc. DCASE*, 2020.

[7] K. Wilkinghoff, "Sub-cluster AdaCos: Learning representations for anomalous sound detection," in *Proc. IJCNN*, 2021.

[8] S. Venkatesh, G. Wichern, A. S. Subramanian, and J. Le Roux, "Improved domain generalization via disentangled multi-task learning in unsupervised anomalous sound detection," in *Proc. DCASE*, 2022.

[9] A. Jiang *et al.*, "AnoPatch: Towards better consistency in machine anomalous sound detection," in *Proc. Interspeech*, 2024.

[10] ——, "Adaptive prototype learning for anomalous sound detection with partially known attributes," in *Proc. ICASSP*, 2025.

[11] A. I. Mezza, G. Zanetti, M. Cobos, and F. Antonacci, "Zero-shot anomalous sound detection in domestic environments using large-scale pretrained audio pattern recognition models," in *Proc. ICASSP*, 2023.

[12] P. Saengthong and T. Shinozaki, "Deep generic representations for domain-generalized anomalous sound detection," in *Proc. ICASSP*, 2025.

[13] H.-H. Wu, W.-C. Lin, A. Kumar, L. Bondi, S. Ghaffarzadegan, and J. P. Bello, "Towards few-shot training-free anomaly sound detection," in *Proc. Interspeech*, 2025.

[14] B. Han, A. Jiang, X. Zheng, W.-Q. Zhang, J. Liu, P. Fan, and Y. Qian, "Exploring self-supervised audio models for generalized anomalous sound detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 33, 2025.

[15] Y. Zhang, J. Liu, and M. Li, "ECHO: Frequency-aware hierarchical encoding for variable-length signal," *arXiv preprint arXiv:2508.14689*, 2025.

[16] P. Fan *et al.*, "FISHER: A foundation model for multi-modal industrial signal comprehensive representation," *arXiv preprint arXiv:2507.16696*, 2025.

[17] K. Wilkinghoff, S. Yadav, and Z.-H. Tan, "Temporal pooling strategies for training-free anomalous sound detection with self-supervised audio embeddings," *arXiv preprint*, 2026.

[18] K. Wilkinghoff, H. Yang, J. Ebbers, F. G. Germain, G. Wichern, and J. Le Roux, "Keeping the balance: Anomaly score calculation for domain generalization," in *Proc. ICASSP*, 2025.

[19] ——, "Local density-based anomaly score normalization for domain generalization," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 33, 2025.

[20] K. Wilkinghoff, T. Fujimura, K. Imoto, J. Le Roux, Z.-H. Tan, and T. Toda, "Handling domain shifts for anomalous sound detection: A review of DCASE-related work," in *Proc. DCASE*, 2025.

[21] T. Fujimura, K. Wilkinghoff, K. Imoto, and T. Toda, "ASDKit: A toolkit for comprehensive evaluation of anomalous sound detection methods," in *Proc. DCASE*, 2025.

[22] M. Matsumoto, T. Fujimura, W. Huang, and T. Toda, "Adjusting bias in anomaly scores via variance minimization for domain-generalized discriminative anomalous sound detection," in *Proc. DCASE*, 2025.

[23] Y. Koizumi *et al.*, "Description and discussion on DCASE2020 Challenge Task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2020.

[24] H. Purohit *et al.*, "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," in *Proc. DCASE*, 2019.

[25] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Proc. WASPAA*, 2019.

[26] K. Dohi *et al.*, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," in *Proc. DCASE*, 2022.

[27] 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*, 2021.

[28] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, and M. Yasuda, "ToyADMOS2+: New Toyadmos data and benchmark results of the first-shot anomalous sound event detection baseline," in *Proc. DCASE*, 2023.

[29] D. Niizumi, N. Harada, Y. Ohishi, D. Takeuchi, and M. Yasuda, "ToyADMOS2#: Yet another dataset for the DCASE2024 challenge task 2 first-shot anomalous sound detection," in *Proc. DCASE*, 2024.

[30] D. Albertini, F. Augusti, K. Esmer, A. Bernardini, and R. Sannino, "IMAD-DS: A dataset for industrial multi-sensor anomaly detection under domain shift conditions," in *Proc. DCASE*, 2024.

[31] N. Harada, D. Niizumi, Y. Ohishi, D. Takeuchi, and M. Yasuda, "ToyADMOS2025: The evaluation dataset for the DCASE2025T2 first-shot unsupervised anomalous sound detection for machine condition monitoring," in *Proc. DCASE*, 2025.

[32] D. K. McClish, "Analyzing a portion of the ROC curve," *Medical decision making*, vol. 9, no. 3, 1989.

[33] K. Wilkinghoff, "Self-supervised learning for anomalous sound detection," in *Proc. ICASSP*, 2024.

[34] ——, "AdaProj: Adaptively scaled angular margin subspace projections for anomalous sound detection with auxiliary classification tasks," in *Proc. DCASE*, 2024.

[35] A. Cramer, H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Proc. ICASSP*, 2019.

[36] S. Chen *et al.*, "BEATs: Audio pre-training with acoustic tokenizers," in *Proc. ICML*, 2023.

[37] J. F. Gemmeke *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017.

[38] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "EAT: self-supervised pre-training with efficient audio transformer," in *Proc. IJCAI*, 2024.

[39] H. Dinkel, Z. Yan, Y. Wang, J. Zhang, Y. Wang, and B. Wang, "Scaling up masked audio encoder learning for general audio classification," in *Proc. Interspeech*, 2024.