APSIPA Transactions on Signal and Information Processing, 2025, 14, e13

Original Paper

Improving Anomalous Sound Detection Through Pseudo-anomalous Set Selection and Pseudo-label Utilization Under Unlabeled Conditions

Ibuki Kuroyanagi[1*], Takuya Fujimura[1], Kazuya Takeda[2] and Tomoki Toda[3]

[1] *The Graduate School of Informatics, Nagoya University, Aichi, Japan*
[2] *The Institutes of Innovation for Future Society, Nagoya University, Aichi, Japan*
[3] *The Information Technology Center, Nagoya University, Aichi, Japan*

ABSTRACT

This paper addresses performance degradation in anomalous sound detection (ASD) when neither sufficiently similar machine data nor operational state labels are available. We present an integrated pipeline that combines three complementary components derived from prior work and extends them to the unlabeled ASD setting. First, we adapt an anomaly-score-based selector to curate external audio data resembling the normal sounds of the target machine. Second, we utilize triplet learning to assign pseudo-labels to unlabeled data, enabling finer classification of operational sounds and detection of subtle anomalies. Third, we employ iterative training to refine both the pseudo-anomalous set selection and pseudo-label assignment, progressively improving detection accuracy. Experiments on the DCASE2022–2024 Task 2 datasets demonstrate that, in unlabeled settings, our approach achieves an average AUC increase of over 6.6 points compared to conventional

*Corresponding author: Ibuki Kuroyanagi, kuroyanagi.ibuki@g.sp.m.is.nagoya-u.ac.jp.
This paper was partly supported by Japan's New Energy and Industrial Technology Development Organization (NEDO), project JPNP20006.

Received 16 March 2025; revised 03 May 2025; accepted 22 May 2025
ISSN 2048-7703; DOI 10.1561/116.20250017

methods. In labeled settings, incorporating external data from the pseudo-anomalous set further boosts performance. These results highlight the practicality and robustness of our methods in scenarios with scarce machine data and labels, facilitating ASD deployment across diverse industrial settings with minimal annotation effort.

Keywords: Anomalous sound detection, pseudo-label, domain shift, external data, triplet learning.

1 Introduction

Anomalous sound detection (ASD) systems assess whether a monitored machine is operating normally or anomalously by placing a microphone nearby and analyzing the captured audio data [2, 3, 23, 29, 31, 20]. Unlike anomaly detection systems relying on images or video [52, 51, 7, 65], ASD systems excel in confined or dark environments inaccessible to humans, detecting anomalies through sound rather than visual inspection. For example, they can identify subtle issues, such as irregularities in high-speed rotating machines or minor wear in excavated areas, that cameras might miss, by capturing acoustic changes.

Developing ASD systems is straightforward when ample anomalous data are available, but real-world deployment faces three key challenges:

1. **Rarity of anomalous data.** Anomalous events are infrequent and diverse, prompting the use of only normal data for training to detect unknown anomalies [31].

2. **Influence of background noise.** Recorded audio comprises normal and anomalous machine sounds mixed with environmental noise, where the distinction between normal and anomalous machine sounds may be less pronounced than that between machine sounds and background noise [62].

3. **Domain shift.** Variations in machine settings and factory environments can cause mismatches between training and testing conditions, leading to misclassification of normal sounds as anomalies [28, 10].

In the literature, unsupervised ASD usually refers to the setting in which only normal recordings are available during training and no anomalous examples are observed. Two primary strategies address these challenges: generative and discriminative model-based methods [28]. Generative methods include two sub-approaches: one minimizes reconstruction errors using autoencoders [53, 17, 48, 27, 42] or generative adversarial networks [63, 26] with normal data, while the other maximizes the likelihood of normal data using normalizing flows [46, 30, 8] or Gaussian mixture models [50, 37, 18]. These methods model the probability density of normal data. Conversely, discriminative methods classify data based on machine type and operational state (*e.g.*, speed, location, microphone type) [38, 58, 34, 5, 24, 66, 19, 6, 14, 61], deriving posterior probabilities for normal states.

Both methods solve challenge 1 (rare anomalies) by training exclusively on normal data, but they diverge on challenges 2 and 3. For challenge 2 (background noise), generative methods must model the entire acoustic scene and thus flag benign noise variations as anomalies, whereas discriminative methods concentrate on machinespecific cues, yielding greater resilience to noise perturbations [62]. Challenge 3 (domain shift), which manifests itself as changes in machine settings or microphone placement, poses difficulties for both methods, since shifts in the distribution of normal sounds degrade detection performance [28, 10]. However, when only a handful of normal samples in the target domain are available, discriminative methods can be adapted by tweaking a few shots to recover performance, while generative methods lack this capability [4]. Extensive benchmarks on DCASE Task 2 and related evaluations confirm these tendencies [44, 59, 33].

However, discriminative methods fail when classification tasks are overly simplistic [31]. If labels are too coarse the network has the potential to capture spurious cues (e.g., noise level or band-limited energy) instead of subtle machine abnormalities. Consequently, performance hinges on access to similar machines and detailed state annotations during training [60]. Such labels are difficult to obtain for new equipment or inaccessible installations [45], leading to high false alarm rates.

Building on, yet going beyond, prior work, we make three integrative contributions: (i) **Pseudo-anomalous set selection**: we extend the anomaly-score selector of [35] by adding machine-specific thresholds and importing the selected AudioSet clips as extra multi-class normals (instead of binary pseudo-anomalies). (ii) **Triplet-based pseudo-label assignment**: we first adopt triplet learning following [15] to improve the fidelity of the generated pseudo-labels; building on this, we introduce a training scheme that leverages these refined pseudo-labels to achieve higher anomaly detection performance. (iii) **Iterative learning:** we show that alternately updating the external set and the pseudo-labels yields consistent AUC gains in DCASE 20222024 Task 2. To confirm the effectiveness of these methods, we conducted extensive experimental validations in unlabeled and labeled settings on DCASE 2022-2024 Task 2 datasets [10, 45, 9].[1]

[1] Our implementation is available at https://github.com/ibkuroyagi/unlabeled-asd.

This paper is structured as follows: Section 2 details discriminative method challenges, Section 3 reviews state-of-the-art labeled approaches, Section 4 presents our method, Section 5 provides experimental results, and Section 6 concludes.

2 The Issues with Discriminative Model-based Methods

Discriminative model-based methods identify differences in operational sounds, such as those caused by machine manufacturers or settings, as defined by labels. When an anomalous sound is input, the model classifies it into a class different from its original class, thereby identifying it as an anomaly. Consequently, these methods are significantly influenced by the types of data in the training set and the granularity of the labels [60].

For example, in the ASD competition DCASE Challenge Task 2, datasets from 2022 to 2024 include operational sounds from 7 to 16 machine types, labeled with machine type, section ID, and attribute. Table 1 shows the details of the datasets. The machine type indicates the kind of machine, the section ID represents domain shifts within the same machine type, and the attribute details the operational settings. Notably, the section ID also serves as a label for product models or manufacturers within certain machine types, making its classification equivalent to distinguishing similar sounds within a machine type. Prior to DCASE2022, methods that classified section IDs to detect subtle differences within machine types achieved high performance [49, 32, 55, 22, 11]. From DCASE2023 onward, when section IDs were unavailable, classifying attribute labels improved performance [45, 9].

Table 1: Details of the datasets used in DCASE Task 2 across different years.

DCASE	# of machine types	# of section IDs	Attribute labels available	# of training samples per machine type
2022	7	6	✓	6000
2023	14	1	✓	1000
2024	16	1	✓	1000

However, label information suitable for ASD, such as section IDs or attributes, is not always accessible. For instance, when deploying an ASD system in a new factory, the monitored machines are often of the same manufacturer and product model [9], making it challenging to collect data equivalent to section IDs. Moreover, obtaining attribute labels is difficult for machines where state monitoring or setting annotation is impractical [45]. Thus, existing methods struggle to achieve high performance when suitable data for ASD are unavailable.

3 State-of-the-art Method Under Labeled Conditions

3.1 Network and Input

We adopt the architecture of Wilkinghoff [60], enhanced with Fujimuras multiresolution refinement [54]. Each audio recording is first converted into three complementary representations: a 2D magnitude spectrogram obtained by applying a 256 mswindow shorttime Fourier transform (STFT), a second 2D magnitude spectrogram using an 8 mswindow STFT, and a 1D magnitude spectrum computed by a discrete Fourier transform (DFT) over the entire signal. Each representation is then fed into its own CNN branch, producing a 128-dimensional embedding $\mathbf{z}^{(m)} \in \mathbb{R}^{128}$, $m = 1, 2, 3$. The three embeddings are concatenated to form $\mathbf{z}^{\text{cat}} = [\, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{z}^{(3)} \,] \in \mathbb{R}^{384}$.

3.2 Sub-cluster AdaCos and Subspace Loss

Following [62], each embedding is trained with the Sub-cluster AdaCos (SCAC) loss, which shrinks intra-class variance and enlarges inter-class margins. Fujimura *et al.* [15] introduce the subspace loss

$$\mathcal{L}_{\text{ss}} = \mathcal{L}_{\text{SCAC}}(\mathbf{z}^{\text{cat}}, l) + \sum_{m=1}^{3} \mathcal{L}_{\text{SCAC}}\big(\mathbf{z}^{(m)}, l\big), \tag{1}$$

where l is the one-hot label that combines machine type and attribute. The first term fixes class centres to stabilise the global embedding, whereas the second terms encourage each sub-space to carry discriminative cues on its own, approximating FeatEx [60] behaviour without explicit feature swapping. Data augmentation uses mixup [67] exactly as reported in [60, 54].

3.3 Inference

After training, source-domain embeddings are clustered by k_{so}-means and the target-domain ones by k_{ta}-means (k_{so}=16, k_{ta}=10).

$$\mathcal{C}_{\text{so}} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{k_{\text{so}}}\}, \tag{2}$$

$$\mathcal{C}_{\text{ta}} = \{\mathbf{c}_{k_{\text{so}}+1}, \ldots, \mathbf{c}_{k_{\text{so}}+k_{\text{ta}}}\}, \tag{3}$$

$$\mathcal{C} = \mathcal{C}_{\text{so}} \cup \mathcal{C}_{\text{ta}}, \quad J = k_{\text{so}} + k_{\text{ta}}. \tag{4}$$

For a test embedding $\mathbf{z}$ we compute the cosine similarity to every representative

$$s_j(\mathbf{z}) = \frac{\langle \mathbf{z}, \mathbf{c}_j \rangle}{\|\mathbf{z}\| \, \|\mathbf{c}_j\|}, \qquad j = 1, \ldots, J. \tag{5}$$

The anomaly score is then

$$\mathrm{score}(\mathbf{z}) = -\max_{1 \leq j \leq J} s_j(\mathbf{z}), \tag{6}$$

so that larger values indicate that $\mathbf{z}$ lies farther from every normal cluster.

4 Proposed Method

In unlabeled conditions where section IDs and attributes are unavailable, the performance of discriminative models significantly degrades [15]. Building upon the subspace loss $\mathcal{L}_{\mathrm{ss}}$ and inference procedure described in Section 3, we propose new steps to address the lack of labels and enhance ASD performance. Our proposed method consists of three components:

1. **Pseudo-anomalous set selection from external data:** External data similar to the normal data of the target machine type are selected based on a machine-specific threshold.
2. **Assigning pseudo-labels to unlabeled data:** Class labels are assigned to unlabeled data using triplet learning.
3. **Iterative learning:** The model is retrained iteratively to refine performance.

An overview of the proposed methods is shown in Figure 1. We describe each component in detail in the following subsections.

4.1 Pseudo-anomalous Set Selection from External Data

The performance of discriminative models decreases when class labels representing similar sounds within the same machine type (*e.g.*, section IDs) are unavailable. The proposed method addresses this by selecting external data that resemble the target machine type's normal sounds and assigning appropriate class labels. An overview of this process is shown in Figure 2. The pseudo-anomalous set selection consists of three steps: (i) training a baseline model, (ii) selecting the pseudo-anomalous set from external data using a machine-specific threshold, and (iii) retraining the baseline model with the pseudo-anomalous set. We adopt the baseline method proposed by Fujimura [54] to train the model for selecting external data.

4.1.1 Selecting the Pseudo-anomalous Set from External Data

To mitigate the performance drop, the proposed method selects external data that are misclassified as normal by the baseline model. The trained baseline

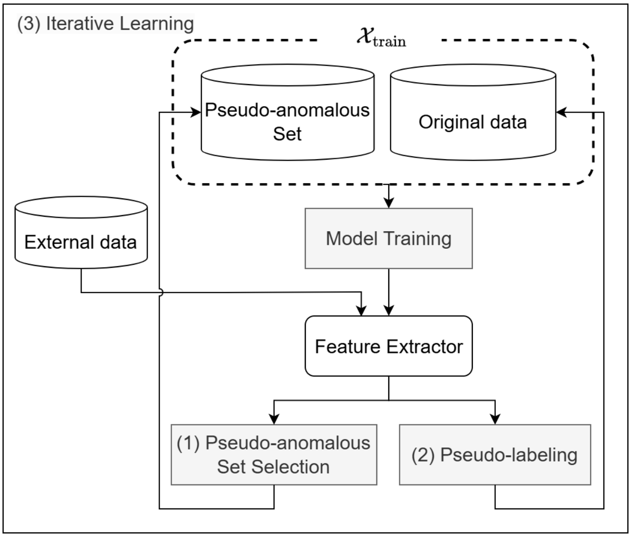

Figure 1: Overview of the proposed method, illustrating the integration of three key components within an iterative learning framework: (1) Selection of the pseudo-anomalous set from external data using a feature extractor, (2) Assignment of pseudo-labels to unlabeled original data via the same feature extractor, and (3) Iterative learning, where the model is retrained over multiple cycles using updated training data $\mathcal{X}_{\text{train}}$ derived from both the pseudo-anomalous set and original data to progressively improve performance.

model computes anomaly scores for all training data, and for each machine type, the highest anomaly score is used as a threshold, denoted as $a^{\text{thr}}_{\text{machine}}$. External data with anomaly scores below $a^{\text{thr}}_{\text{machine}}$ are considered similar to the normal data of the corresponding machine type. To prevent over-reliance on external data, the number of external samples added per machine type is capped at N_{max}. Specifically, if N_{out} is the number of selected external samples, the number of samples added is:

$$N_{\text{ex}} = \min(N_{\text{out}}, N_{\text{max}}), \tag{7}$$

where the N_{ex} external samples with the smallest anomaly scores are added to the training data. This expanded dataset is referred to as $\mathcal{X}_{\text{train}}$. The classification labels for the external data follow the format `machine_attribute`, where:

- `machine` is assigned based on the machine type from the original dataset, using the class with the highest similarity to the external dataset.
- `attribute` is assigned from the external dataset's class label (if available).

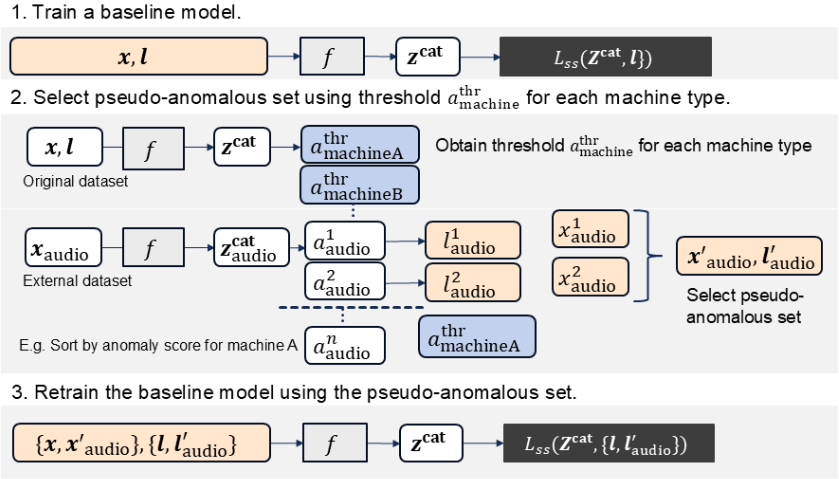
1. Train a baseline model.
$\boldsymbol{x}, \boldsymbol{l}$
f
$\boldsymbol{z}^{\text{cat}}$
$L_{ss}(\boldsymbol{Z}^{\text{cat}}, \boldsymbol{l}\})$
2. Select pseudo-anomalous set using threshold $a^{\text{thr}}_{\text{machine}}$ for each machine type.
$\boldsymbol{x}, \boldsymbol{l}$
f
$\boldsymbol{z}^{\text{cat}}$
$a^{\text{thr}}_{\text{machineA}}$
$a^{\text{thr}}_{\text{machineB}}$
Obtain threshold $a^{\text{thr}}_{\text{machine}}$ for each machine type
Original dataset
$\boldsymbol{x}_{\text{audio}}$
f
$\boldsymbol{z}^{\text{cat}}_{\text{audio}}$
a^{1}_{audio}
a^{2}_{audio}
l^{1}_{audio}
l^{2}_{audio}
x^{1}_{audio}
x^{2}_{audio}
$\boldsymbol{x}'_{\text{audio}}, \boldsymbol{l}'_{\text{audio}}$
External dataset
Select pseudo-anomalous set
E.g. Sort by anomaly score for machine A
a^{n}_{audio}
$a^{\text{thr}}_{\text{machineA}}$
3. Retrain the baseline model using the pseudo-anomalous set.
$\{\boldsymbol{x}, \boldsymbol{x}'_{\text{audio}}\}, \{\boldsymbol{l}, \boldsymbol{l}'_{\text{audio}}\}$
f
$\boldsymbol{z}^{\text{cat}}$
$L_{ss}(\boldsymbol{Z}^{\text{cat}}, \{\boldsymbol{l}, \boldsymbol{l}'_{\text{audio}}\})$

Figure 2: Overview of the process for selecting the pseudo-anomalous set from external data, consisting of three steps: (1) Training a baseline model on the original dataset with the subspace loss $\mathcal{L}_{\text{ss}}$, (2) Processing external data through the baseline model to compute anomaly scores, which are then sorted and filtered using machine-specific thresholds $a^{\text{thr}}_{\text{machineA}}, a^{\text{thr}}_{\text{machineB}}, \ldots$ to select the pseudo-anomalous set, and (3) Retraining the baseline model on the combined dataset to refine predictions with the subspace loss $\mathcal{L}_{\text{ss}}$.

If multiple external samples belong to the same class but are associated with different machine types, they are treated as separate classes to account for the potentially coarser class definitions in external data.

Conventional methods using external data [35, 47] randomly select external data, define them as pseudo-anomalous, and train the model using binary classification. However, these methods may select irrelevant sounds (*e.g.*, instruments or speech), limiting their effectiveness. Our method improves performance by filtering external data that are beneficial for ASD and treating them as part of a multi-class classification problem, enabling the detection of subtle differences among normal data.

4.1.2 Retraining the Model with the Pseudo-anomalous Set

The feature extractor is retrained on the augmented dataset $\mathcal{X}_{\text{train}}$ using the same procedure as the baseline. After training, representative vectors are calculated only from the original dataset, excluding external data, to prevent misclassifying anomalous sounds similar to external data as normal. The methods for calculating anomaly scores and representative vectors remain the same as in the baseline [54].

4.2 Assigning Pseudo-labels to Unlabeled Data

We propose a method to improve performance when the machine type is known, but internal state or configuration labels (*e.g.*, `attribute`) are unavailable. An overview of this method is shown in Figure 3. The process consists of three steps: (i) training a baseline model, (ii) obtaining pseudo-labels from the baseline model, and (iii) retraining the baseline model with the pseudo-labels.

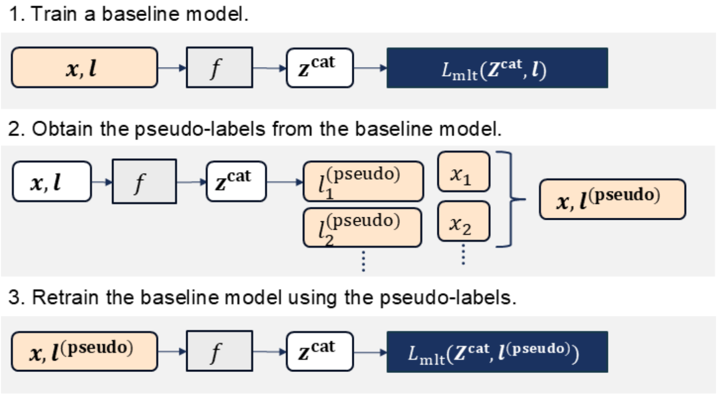
1. Train a baseline model.
x, l
f
z^{cat}
$L_{\text{mlt}}(Z^{\text{cat}}, l)$
2. Obtain the pseudo-labels from the baseline model.
x, l
f
z^{cat}
$l_1^{(\text{pseudo})}$
$l_2^{(\text{pseudo})}$
x_1
x_2
$x, l^{(\text{pseudo})}$
3. Retrain the baseline model using the pseudo-labels.
$x, l^{(\text{pseudo})}$
f
z^{cat}
$L_{\text{mlt}}(Z^{\text{cat}}, l^{(\text{pseudo})})$

Figure 3: Overview of the proposed method for assigning pseudo-labels to unlabeled data, consisting of three steps: (1) Training a baseline model on the original dataset by passing it through the function f to produce category outputs z^{cat}, optimized using the loss function $\mathcal{L}_{\text{mlt}}$, (2) Using the trained baseline model to predict and assign pseudo-labels to unlabeled data, generating $x, l^{(\text{pseudo})}$, and (3) Retraining the model on the pseudo-labeled data $x, l^{(\text{pseudo})}$ to iteratively improve predictions via the loss function $\mathcal{L}_{\text{mlt}}$.

4.2.1 Training a Baseline Model

In unlabeled conditions where attribute labels are unavailable, baseline training typically relies solely on machine type labels. However, this approach often leads to a simplistic classification task, causing the model to focus on irrelevant features such as specific frequencies or background noise [62], which can impair its ability to detect subtle anomalies. To address this limitation and prepare the model for effective pseudo-labeling, we enhance the baseline training by incorporating triplet learning, building on insights from prior work [15]. Specifically, [15] demonstrated that triplet learning can effectively disentangle operational sound variations from environmental noise, creating a feature space better suited for anomaly detection. We selected triplet learning instead of contrastive learning because it better aligns with our goal of capturing subtle changes within the same sample, such as variations in machine operational sounds, rather than just distinguishing between different samples. Contrastive learning focuses on separating distinct samples within a mini-batch, which is less effective for identifying fine-grained differences critical to anomaly detection. In contrast, triplet learning encourages the model to emphasize these subtle intra-sample changes while ignoring irrelevant noise variations.

For triplet learning, we define three samples: the anchor $\boldsymbol{x}_i^a$, the positive $\boldsymbol{x}_i^p$, and the negative $\boldsymbol{x}_i^n$, as follows:

- **Anchor**: The anchor sample $\boldsymbol{x}_i^a$ is a normal sound sample from the i-th machine type.

- **Positive**: The positive sample $\boldsymbol{x}_i^p$ is created by adding scaled sound from a different machine type $j \neq i$ as background noise to the anchor, defined as:
$$\boldsymbol{x}_i^p = \boldsymbol{x}_i^a + 10^{-\frac{\alpha}{20}} \cdot \frac{\|\boldsymbol{x}_i^a\|}{\|\boldsymbol{x}_j\|}\boldsymbol{x}_j, \tag{8}$$
where $\|\cdot\|$ denotes the Euclidean norm, and α is a hyperparameter representing the signal-to-noise ratio (SNR) in decibels to adjust the intensity of the background noise relative to the anchor signal.

- **Negative**: The negative sample $\boldsymbol{x}_i^n$ is generated by pitch-shifting the anchor sample:
$$\boldsymbol{x}_i^n = \mathrm{PitchShift}(\boldsymbol{x}_i^a, \beta), \tag{9}$$
where β simulates operational variations, with the implementation based on torchaudio [64, 25].

This triplet configuration encourages the model to distinguish operational changes from background noise variations. Let $\boldsymbol{z}_i^a$, $\boldsymbol{z}_i^p$, and $\boldsymbol{z}_i^n$ represent the embedding vectors of $\boldsymbol{x}_i^a$, $\boldsymbol{x}_i^p$, and $\boldsymbol{x}_i^n$, respectively.

A similarity function with a temperature parameter τ is employed:

$$s_\tau(\boldsymbol{z}, \boldsymbol{z}') = \frac{\langle \boldsymbol{z}, \boldsymbol{z}' \rangle}{\tau \|\boldsymbol{z}\| \|\boldsymbol{z}'\|}, \tag{10}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product. The triplet loss $\mathcal{L}_{\mathrm{trp}}$ is defined as:

$$\mathcal{L}_{\mathrm{trp}}(\boldsymbol{z}_i^a, \boldsymbol{z}_i^p, \boldsymbol{z}_i^n) = \max\left\{0, \gamma + 1 - s_\tau(\boldsymbol{z}_i^a, \boldsymbol{z}_i^p) + s_\tau(\boldsymbol{z}_i^a, \boldsymbol{z}_i^n)\right\}, \tag{11}$$

where γ is a margin parameter. This loss encourages the model to prioritize operational sound differences over noise.

The feature extractor is trained using both the triplet loss $\mathcal{L}_{\mathrm{trp}}$ and the subspace loss $\mathcal{L}_{\mathrm{ss}}$, where $\boldsymbol{l}_i$ is the one-hot machine type label. Mixup [67] is applied to $\mathcal{L}_{\text{ss}}$ with a 50 % probability, but not to $\mathcal{L}_{\text{trp}}$, to preserve the intended triplet relationships. The combined training loss is:

$$\mathcal{L}_{\text{mlt}} = \mathcal{L}_{\text{trp}}(\boldsymbol{z}_i^a, \boldsymbol{z}_i^p, \boldsymbol{z}_i^n) + \mathcal{L}_{\text{ss}}(\boldsymbol{z}_i, \boldsymbol{l}_i). \tag{12}$$

This enhanced baseline training establishes a robust feature space, enabling effective pseudo-label assignment in the subsequent steps and supporting the iterative learning framework.

4.2.2 Generating Pseudo-labels

After training, we generate pseudo-labels by applying k-means clustering to the learned embeddings for each domain and machine type, similar to previous studies [15, 57, 40]. Let k_{so} and k_{ta} be the numbers of clusters for the source and target domains, respectively. Each sample $\boldsymbol{x}_i$ is assigned to the cluster with the nearest centroid in embedding space:

$$\ell_i^{(\text{pseudo})} = \mathop{\arg\min}_{1 \leq j \leq k_{d_i}} \left\| \boldsymbol{z}_i - \boldsymbol{c}_j^{(d_i)} \right\|, \quad \text{where } d_i = \begin{cases} \text{so} & \text{(source domain)}, \\ \text{ta} & \text{(target domain)}. \end{cases} \tag{13}$$

4.2.3 Retraining the Baseline Model with Pseudo-labels

We retrain the model from scratch using $\mathcal{L}_{\text{mlt}}$, replacing $\boldsymbol{l}_i$ with the pseudo-label $\ell_i^{(\text{pseudo})}$. Since pseudo-labels are not ground truth, minimizing intra-class variance with $\mathcal{L}_{\text{ss}}$ alone may force samples with different true labels into the same cluster, leading the model to focus on environmental noise or incorrectly group operational sounds. By incorporating the triplet loss, the model learns to ignore noise differences and focus on operational sound changes, mitigating these issues and improving ASD performance. For inference, the procedure remains the same as the baseline.

4.3 Iterative Selection of Pseudo-anomalous Set and Pseudo-labels

We combine the pseudo-anomalous set selection (Section 4.1) and pseudo-label assignment (Section 4.2) into an iterative training scheme:

- **Stage 1:** Train the model using $\mathcal{L}_{\text{mlt}}$ on the original labeled dataset (no external data or pseudo-labels).
- **Stage M ($M \geq 2$):**
  1. Use the model from stage ($M-1$) to select the pseudo-anomalous set and add up to $N_{\max}$ samples per machine type to the training set.

2. Use the same model to assign pseudo-labels via k-means clustering.
3. Retrain the model using the augmented dataset and new pseudo-labels with $\mathcal{L}_{\text{mlt}}$.

An overview of this method is shown in Figure 1. By iterating, we refine the model at each stage, potentially obtaining more accurate pseudo-anomalous sets and meaningful pseudo-labels for the next stage. During inference, we reuse the procedure summarised in Section 3: for each test embedding we take the negative of its maximum cosine similarity to the representative vectors obtained from source and target domain training data. This iterative approach progressively enhances performance by leveraging both external data and pseudo-labeled internal variations.

5 Experimental Evaluations

5.1 Datasets

We evaluated our proposed methods using the DCASE Task 2 datasets from 2022 to 2024 [10, 45, 9]. These datasets comprise machine sound recordings, including factory background noise. Each recording is a single-channel audio file, lasting 6–18 seconds, with a 16 kHz sampling rate. The DCASE2022 dataset includes seven machine types: fan, gearbox, bearing, slide rail (slider), valve, ToyCar, and ToyTrain [22, 12]. The DCASE2023 dataset features 14 machine types: its development set matches DCASE2022, while the evaluation set includes ToyDrone, ToyTank, ToyNscale, bandsaw, grinder, and shaker [21]. The DCASE2024 dataset contains 16 machine types: its development set aligns with DCASE2022, and the evaluation set includes 3D-Printer, AirCompressor, BrushlessMotor, HairDryer, HoveringDrone, RoboticArm, Scanner, ToothBrush, and ToyCircuit [1, 43]. Table 1 summarizes the label conditions for each dataset. Note that some attribute labels in DCASE2024, unavailable during the competition, were released post-competition and used in this analysis.

Each dataset provides 1,000 training samples per machine type, all normal data. These consist of 990 source-domain samples and 10 target-domain samples affected by domain shift. Domain shift occurs due to changes in machine sound characteristics, such as those caused by maintenance actions or variations in the acoustic environment (*e.g.*, background noise or shifts in operating conditions). Training samples include attribute labels indicating the machine's operating state or environment. An ideal ASD system should detect anomalies reliably despite domain shifts without adaptation [56]. Per the DCASE Task 2 setup, training data indicate source or target domain origin. Evaluation data include 100 normal and 100 anomalous samples per machine type, split evenly between domains. During inference, the domain of test data is unknown. In DCASE2022, section IDs denote domain shift types, used alongside attribute labels in labeled settings. From DCASE2023 onward, the absence of section IDs simplified the classification task, making it harder to extract embeddings sensitive to anomalous changes.

Performance is assessed using the area under the ROC curve (AUC), following DCASE Task 2. The AUC is vital as machine condition monitoring thresholds aim to minimize false alarms [41, 13]. This threshold-independent metric offers an objective comparison of ASD systems.

5.2 System Descriptions

In this study, we compare our proposed method in an unlabeled configuration with Wilkinghoff's method, known for state-of-the-art performance across multiple datasets in conventional ASD systems [60], and Fujimura's baseline method [54]. Since ASD hyperparameters cannot be tuned with anomalous data, we use identical hyperparameter values for all machine types. This approach, widely adopted in ASD [10, 45, 9], ensures robust performance on unseen machine types. We outlined the settings for each method below.

Wilkinghoff's Method (Wilkinghoff [60]). This method, per [60], uses a magnitude spectrogram and the full magnitude spectrum as input features. Feature extraction employs a window size of 64 ms with a 50 % hop size. Two convolutional branches produce 128-dimensional embedding vectors, concatenated into a 256-dimensional feature for ASD. The SCAC loss [62] is applied to 16 untrainable sub-clusters (randomly initialized) using these 256-dimensional features. Training involved a batch size of 100, 50 epochs, the AdamW optimizer [39] with a learning rate of 0.001, and mixup with uniformly sampled ratios. Post-training, embeddings are clustered via k-means: 16 clusters for the source domain and 10 for the target domain, where each target-domain sample serves as its own representative vector. The optimal cluster number in DCASE2023 was 16 for the source domain, though its impact was minor [60]. Anomaly scores are derived from cosine similarity between test samples and representative vectors.

Baseline (`Ba` [54]). This baseline adopts most settings from Wilkinghoff [60], with key differences: (i) it uses two magnitude spectrograms (8 ms and 256 ms windows) plus the full magnitude spectrum, all with 50 % hop sizes; (ii) three convolutional branches each yield a 128-dimensional embedding, concatenated to form a 384-dimensional feature; (iii) SCAC loss [62] is applied twice, first to 16 untrainable sub-clusters for the 384-dimensional feature, then to 16 trainable sub-clusters per 128-dimensional branch (all randomly initialized). Other training and inference parameters align with Wilkinghoff's method.

Baseline with External Data (`Ba+Ex`). Following Section 4.1, we incorporate approximately 1.8 million labeled audio samples from Audioset [16] as external data. Each Audioset sample carries a “mid” label (*e.g.*, `/m/05r5c` for Piano, `/m/05zppz` for Male speech), used as attribute labels. Representative vectors are computed from the training set using k-means with 16 clusters for the source domain. Up to $N_{\text{max}} = 1000$ external samples per machine type, matching the training data size, are selected. The baseline model is retrained with these labeled samples, retaining the original hyperparameters.

Baseline with $\mathcal{L}_{\text{trp}}$ (`Ba`+$\mathcal{L}_{\text{trp}}$). Following Section 4.2.1, this method enhances the baseline with triplet learning. Positive samples are generated by adding scaled sound from a different machine type as background noise, with the hyperparameter α (SNR, $[-5, 20]$ dB) controlling noise intensity. The range of $[-5, 20]$ dB was chosen to balance the intensity of the noise and the original sound source, ensuring that the noise-to-signal ratio remains within a range where both components are comparable. Negative samples are created by pitch shifting the anchor, with β ranging from ±6 to ±12 semitones; this implementation is based on torchaudio [64, 25]. The range of β was selected to simulate realistic operational variations without excessively distorting the original sound, covering a semitone shift to an octave shift. The triplet loss uses $\tau = 0.2$ and $\gamma = 0.5$, following the configuration in [36].

Baseline with Pseudo-Labels (`Ba`+`Ps`). Per Section 4.2.2, this method retrains the baseline using pseudo-labels. After initial training, embeddings are clustered per domain: 16 clusters ($k_{\text{so}} = 16$) for the source domain and 4 ($k_{\text{ta}} = 4$) for the target domain per machine type. Assuming 16 source-domain attributes and 4 target-domain attributes, this yields 20 pseudo-classes per machine type. Samples are assigned pseudo-labels based on the nearest of 20 centroids. From stage 2, the model will address a classification task with 20 times the classes of stage 1’s machine types.

Iterative Learning Method (`Ba`+$\mathcal{L}_{\text{trp}}$+`Ps`+`Ex`, Stage M). Per Section 4.3, this iterative method boosts performance over up to $M = 5$ iterations. From stage $M = 3, 4, 5$, pseudo-labels and external data are derived using the model from stage $M - 1$.

5.3 Performance Evaluation When Labels are Unavailable

Table 2 presents the experimental results evaluating the performance of the proposed methods under unlabeled conditions. The per-domain performance is reported in Table A.1. To assess the effect of $\mathcal{L}_{\text{trp}}$, we compare `Ba` and `Ba`+$\mathcal{L}_{\text{trp}}$, focusing on the “all” column. Incorporating $\mathcal{L}_{\text{trp}}$ improved performance on all datasets except DCASE2024 development, where the difference was minimal. This suggests that $\mathcal{L}_{\text{trp}}$ generally enhances performance, likely by suppressing noise and emphasizing variations in operating sounds within samples, creating a more discriminative feature space.

To verify the effect of retraining with external data, we compared `Ba`+$\mathcal{L}_{\text{trp}}$ and `Ba`+$\mathcal{L}_{\text{trp}}$+`Ex`, focusing on the “all” column. Using external data improved

Table 2: Average AUC (%) for each training configuration on the DCASE 2022–2024 Task 2 datasets. The upper block (“w/ label”) is a label-based reference, while the lower block (“w/o label”) shows the proposed unlabeled configurations. Column “stage indicates the iteration number: stage 1 is the initial model, stage 2 adds external data and / or pseudo-labels derived from stage 1, and stages 35 repeat the same update procedure recursively. Columns “dev and “eval correspond to development and evaluation splits. Values are mean ± variance over five random seeds. Ba = baseline [54], Ex = selected external data, Ps = pseudo-labels, and $\mathcal{L}_{\text{trp}}$ = triplet loss.

Use label	Method	stage	2022		2023		2024	
			dev	eval	dev	eval	dev	eval
w/ label	Wilkinghoff [60]	1	82.5±0.8	73.1±0.9	67.2±0.8	74.2±0.3	72.6±0.7	61.5±0.6
	Ba [54]	1	81.9±0.9	73.0±0.3	70.5±0.5	77.5±0.4	72.8±0.4	63.2±0.8
w/o label	Wilkinghoff [60]	1	67.2±5.8	64.1±0.8	62.5±0.8	64.4±3.2	59.7±1.1	55.6±0.6
	Ba [54]	1	71.3±0.9	64.8±0.7	64.2±1.2	67.8±1.4	59.5±0.7	53.8±0.6
	Ba+$\mathcal{L}_{\text{trp}}$	1	71.8±1.6	65.2±1.3	64.1±1.2	68.8±0.6	59.7±1.3	54.1±1.5
	Ba+$\mathcal{L}_{\text{trp}}$+Ex	2	74.0±0.5	65.3±1.3	65.0±1.1	69.2±0.3	61.2±1.3	54.9±0.9
	Ba+$\mathcal{L}_{\text{trp}}$+Ps	2	76.2±0.4	68.4±0.8	64.2±1.3	72.4±0.7	65.5±1.5	56.4±1.1
	Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	2	75.8±0.9	69.1±0.7	64.4±0.5	72.7±1.0	66.4±2.2	56.5±0.6
	Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	3	76.8±1.7	70.1±1.4	**65.2±0.5**	72.6±1.3	68.3±1.0	**57.0±0.1**
	Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	4	76.5±2.0	70.1±0.6	64.3±1.9	**73.1±0.6**	**70.7±1.5**	56.1±0.0
	Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	5	**78.1±1.0**	**70.3±1.1**	**65.2±1.4**	72.6±0.3	70.4±1.5	56.8±0.1

performance across all datasets, demonstrating its effectiveness. Specifically, adding external data similar to normal data appears to enhance the detection of anomalies resembling those external samples.

To examine the effect of retraining with pseudo-labels, we compared Ba+$\mathcal{L}_{\text{trp}}$ and Ba+$\mathcal{L}_{\text{trp}}$+Ps, focusing on the “all” column. Retraining with pseudo-labels improved performance on all datasets, indicating its effectiveness. Assigning samples with similar operating sounds to the same class enables the detection of finer differences, such as those across machine settings, leading to performance gains.

To determine whether combining external data and pseudo-labels yields additional benefits, we compared Ba+$\mathcal{L}_{\text{trp}}$+Ex, Ba+$\mathcal{L}_{\text{trp}}$+Ps, and Ba+$\mathcal{L}_{\text{trp}}$+Ps +Ex, focusing on the “all” column. In most datasets, the combined method outperformed each individual method, except in DCASE2022 development (where Ba+$\mathcal{L}_{\text{trp}}$+Ps performed best) and DCASE2024 development (where Ba+$\mathcal{L}_{\text{trp}}$ +Ex performed best). In these exceptions, target-domain performance was notably high, suggesting that particularly suitable external data or pseudo-labels were obtained. Since combining both methods never degraded performance relative to each single method across all datasets, we consider their joint use effective.

We next evaluated the performance improvements from iterative learning by comparing Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex at stages 2, 3, 4, and 5, focusing on the “all” column. The average scores across all datasets increased from 67.5 at stage 2 to 68.3, 68.5, and 68.9 at stages 3, 4, and 5, respectively, indicating steady improvement. Performance improved in both source and target domains, demonstrating that the proposed method effectively boosts performance under unlabeled conditions. For individual datasets, stages 3, 4, and 5 outperformed stage 2, but differences among stages 3, 4, and 5 were not pronounced. Given that anomalous data are unavailable for model validation, iterating up to stage 3 appears sufficient for developing ASD systems for unknown machines.

Finally, we compared our unlabeled results with the upper-bound performance using ground-truth labels. The average "all"-column scores across all datasets for w/ label Wilkinghoff [60], w/ label `Ba` [54], w/o label Wilkinghoff [60], and w/o label `Ba` [54] are 71.9, 73.2, 62.3, and 63.6, respectively. In `Ba` [54], the gap between labeled and unlabeled settings was 9.4 points, but our approach at stage 5 reduced this gap to 4.3 points, improving performance by 5.1 points. Moreover, our stage 5 result is only 3.0 points below the w/ label Wilkinghoff [60] score. This significant reduction in the performance gap shows that our method greatly improves upon existing unlabeled approaches and closely approaches label-based performance.

These results indicate that, in an unlabeled setting, using $\mathcal{L}_{\text{trp}}$, external data, and pseudo-labeling with at least three iterations is effective.

5.4 Performance Evaluation When Labels are Available

We evaluated the proposed method in a labeled setting using the results presented in Table 3. The per-domain performance is reported in Table A.2. The pseudo-label approach assumes that original labels lack sufficient granularity for ASD. It augments these labels with pseudo-labels to enable finer-grained classification. Specifically, when applying pseudo-labels, the label format shifts from `machine_attribute` to `machine_attribute_pseudo-label`.

Table 3: Average AUC (%) of each supervised configuration (attribute labels available) on the DCASE 20222024 Task 2 datasets. Column "stage indicates the training iteration: stage 1 is the initial model; stage 2 retrains the baseline with external data (Ex) and / or pseudo-labels (Ps) derived from stage 1; stage 3 repeats the same update. Columns "dev and "eval correspond to the development and evaluation splits, respectively (mean ± variance over five random seeds). `Ba` = baseline [54], $\mathcal{L}_{\text{trp}}$ = triplet loss.

Method	stage	2022		2023		2024	
		dev	eval	dev	eval	dev	eval
Wilkinghoff [60]	1	**82.5±0.8**	74.2±0.3	73.1±0.9	72.6±0.7	67.2±0.8	61.5±0.6
`Ba` [54]	1	81.9±0.9	**77.5±0.4**	73.0±0.3	72.8±0.4	70.5±0.5	63.2±0.8
`Ba`+$\mathcal{L}_{\text{trp}}$	1	80.7±0.8	68.8±0.3	72.6±2.3	69.0±0.9	68.8±0.9	61.9±1.6
`Ba`+`Ex`	2	81.7±0.3	76.9±0.5	**73.8±1.0**	**75.0±0.9**	**71.2±0.9**	**64.4±0.6**
`Ba`+`Ps`	2	79.8±0.4	76.0±0.4	73.6±0.6	71.7±1.6	70.2±0.6	61.0±1.1
`Ba`+`Ps`+$\mathcal{L}_{\text{trp}}$	2	80.0±0.5	75.8±0.5	72.5±0.6	69.0±1.6	69.1±0.9	60.7±0.8
`Ba`+`Ex`	3	82.2±0.8	76.8±0.5	73.4±0.5	**75.0±1.5**	70.2±1.1	**64.4±1.2**

We compared `Ba` [54] and `Ba`+$\mathcal{L}_{\text{trp}}$ to evaluate triplet learning's impact. When ground-truth labels are available, incorporating $\mathcal{L}_{\text{trp}}$ degrades performance across all datasets. Since $\mathcal{L}_{\text{trp}}$ drives the model to detect subtle intra-sample variations, it may overemphasize minor sound differences when original labels already adequately represent machine settings. This oversensitivity likely explains the performance decline.

Next, we examined the effect of external data by comparing `Ba` [54] with `Ba+Ex` in the "all" column. Given that $\mathcal{L}_{\mathrm{trp}}$ does not enhance performance in this setting, we use `Ba` [54] as the stage 1 model. Adding external data improves performance across all DCASE2023 and DCASE2024 datasets. However, no significant improvement occurs in DCASE2022. When using external data, samples similar to the normal data are selected from external sources. In settings without predefined machine type distinctions (*e.g.*, unlabeled conditions) or similar machine sounds (*e.g.*, lacking section IDs), external data increase classification complexity, potentially boosting performance. Conversely, when section IDs are present, the training data already contain ample similar operational sounds, limiting the impact of external data on task complexity and thus yielding minimal gains.

We then evaluated pseudo-labeling by comparing `Ba` [54] with `Ba+Ps` in the "all" column. Pseudo-labels degrade performance in all datasets except the DCASE2023 development set. In a labeled setting, pseudo-labeling effectively subdivides existing labels into finer categories. If original labels accurately reflect machine settings, further subdivision fragments the data unnecessarily. These additional categories may capture irrelevant variations (*e.g.*, background noise), rendering pseudo-labeling redundant when the original data are well-segmented.

Finally, we compared stage 2 and stage 3 of `Ba+Ex` to assess iterative learning benefits. Since neither $\mathcal{L}_{\mathrm{trp}}$ nor pseudo-labeling proves advantageous in this setting, we iterate using `Ba+Ex`. Results indicate negligible improvement from stage 2 to stage 3. In stage 2, the model learns to distinguish Audioset samples from normal data based on stage 1 selections. Without new insights from $\mathcal{L}_{\mathrm{trp}}$ or pseudo-labeling, re-extracting external data from Audioset offers no additional perspective, leaving performance largely unchanged.

These findings suggest that in the labeled setting where the original labels are appropriately annotated, applying the external data approach once is the most effective strategy.

5.5 *Effectiveness of Proposed External Data Selection and Impact of External Data Volume*

We investigated the effect of pseudo-anomalous data from external data in the unlabeled settings. Table 4 is analyzed in three key aspects:

1. Performance differences between selecting external data via anomaly scores (our method) and random selection,

Table 4: Performance evaluation comparing external data selection via anomaly scores (proposed method) and random selection, with varying maximum numbers of external data ($N_{\max}$). *Large* N_{out} *machines* refers to machine types where $N_{\text{out}} \geq N_{\max}$, while *small* N_{out} *machines* refers to those where $N_{\text{out}} < N_{\max}$. Each value represents the average AUC [%] across all machine types in the dataset, with variance computed from five runs using different random seeds.

	$N_{\max}$	*large* N_{out} *machines*			*small* N_{out} *machines*		
		2022	2023	2024	2022	2023	2024
`Ba` [54]	–	71.8±1.2	58.4±0.6	59.9±2.3	67.3±0.7	65.6±0.5	59.2±0.6
`Ba+Ex`	500	74.1±1.2	61.2±1.9	65.1±2.7	68.3±0.6	**66.6±0.8**	59.3±0.8
`Ba+Ex`	1000	**74.6±0.8**	**62.3±2.2**	**66.1±5.7**	68.7±1.1	66.3±1.5	**60.0±0.8**
`Ba+Ex`	2000	74.1±0.9	60.6±2.5	63.8±5.4	**69.0±0.6**	66.3±0.6	**60.0±1.0**
`Ba+Ex` (random)	500	73.3±0.9	60.3±2.4	63.6±2.4	67.8±0.9	66.1±1.2	59.8±0.8
`Ba+Ex` (random)	1000	73.5±1.1	60.6±1.9	59.2±2.9	67.4±0.3	65.8±1.0	59.3±0.6
`Ba+Ex` (random)	2000	74.1±0.7	61.5±1.5	61.1±1.1	67.6±0.5	65.1±0.5	58.9±1.5

2. Performance variations based on the maximum number of external samples, $N_{\max}$, used in training,
3. Performance differences between machine types with $N_{\text{out}} \geq N_{\max}$ (*large* N_{out} *machines*) and those with $N_{\text{out}} < N_{\max}$ (*small* N_{out} *machines*).

In DCASE2022, *large* N_{out} *machines* are fan and valve, while *small* N_{out} *machines* are bearing, gearbox, slider, ToyCar, and ToyTrain. In DCASE2023, *large* N_{out} *machines* are bandsaw and grinder, and *small* N_{out} *machines* include bearing, fan, gearbox, shaker, slider, ToyCar, ToyDrone, ToyNscale, ToyTank, ToyTrain, Vacuum, and valve. For DCASE2024, *large* N_{out} *machines* are BrushlessMotor, and *small* N_{out} *machines* are 3DPrinter, AirCompressor, bearing, fan, gearbox, HairDryer, HoveringDrone, RoboticArm, Scanner, slider, ToothBrush, ToyCar, ToyCircuit, ToyTrain, and valve.

To compare our method with random selection, we evaluated `Ba` [54] against `Ba+Ex`. `Ba+Ex` consistently improves performance across all $N_{\max}$ values, whereas `Ba+Ex` (random) shows performance drops in some datasets compared to `Ba` [54]. Notably, the highest performance across all datasets is achieved with our proposed method rather than random selection. This suggests that targeting external samples likely to be misclassified as normal enhances ASD performance more effectively than random selection. Furthermore, this improvement occurs regardless of the classification into *large* N_{out} *machines* or *small* N_{out} *machines*, indicating that adding external data resembling not only the target machine but also co-trained machine types contributes to the performance gain.

Next, we examined the relationship between $N_{\max}$ and performance in `Ba+Ex`. For *large* N_{out} *machines*, performance decreases when $N_{\max}$ is changed from 1000 to either 500 or 2000. This indicates that while incorporating more external data prone to misclassification as normal can improve performance, excessive external data beyond the original training data leads to performance degradation. This highlights the need to balance the volume of external data with the original training data. For *small* N_{out} *machines*, performance shows no consistent trend with varying $N_{\max}$. Since $N_{\text{out}} < N_{\max}$, increasing $N_{\max}$ introduces external data resembling other machine types rather than the target class, which likely explains the lack of clear performance shifts.

These results demonstrate that our method, which prioritizes external data prone to misclassification as normal, outperforms random selection and effectively boosts performance, particularly for machine types with abundant relevant external data (*large* N_{out} *machines*). Furthermore, for these machine types, setting $N_{\max}$ to a value comparable to the size of the training data (*e.g.*, $N_{\max} = 1000$) is critical to prevent performance degradation due to excessive external data.

5.6 Performance Analysis of Triplet Loss and Pseudo-Labels in Stage 2

Table 5 evaluates the performance of models trained in stage 2 using pseudo-labels generated in stage 1 by `Ba` or `Ba` + $\mathcal{L}_{\text{trp}}$. The results demonstrate that pseudo-labels generated by `Ba` + $\mathcal{L}_{\text{trp}}$ in stage 1 consistently yield higher performance than those generated by `Ba` alone, regardless of the loss function used in stage 2. This suggests that incorporating $\mathcal{L}_{\text{trp}}$ in stage 1 enhances the quality of pseudo-labels.

Table 5: Evaluation of the impact of the triplet loss $\mathcal{L}_{\text{trp}}$ on pseudo-label quality and model performance in a two-stage training framework. Columns indicate the model used in stage 1 to generate pseudo-labels Ba, Ba + $\mathcal{L}_{\text{trp}}$ and the dataset (DCASE2022, DCASE2023, DCASE2024). Rows show the model configuration in stage 2: Ba + Ps and Ba + $\mathcal{L}_{\text{trp}}$ + Ps . Each entry reports the average AUC (%) across all machine types in the respective dataset, with mean and variance computed from five runs.

Loss functions in stage 1	`Ba`			`Ba`+$\mathcal{L}_{\text{trp}}$		
DCASE	2022	2023	2024	2022	2023	2024
`Ba+Ps`	70.6±0.2	63.2±1.1	57.4±1.2	71.3±0.5	63.5±1.3	58.9±0.6
`Ba`+$\mathcal{L}_{\text{trp}}$+`Ps`	**71.8±0.4**	**66.8±1.4**	**59.1±0.5**	**74.3±0.5**	**67.0±1.0**	**60.3±1.2**

Moreover, when comparing stage 2 configurations, `Ba` + $\mathcal{L}_{\text{trp}}$ + Ps outperforms `Ba` + Ps across all datasets and pseudo-label data. This indicates that adding $\mathcal{L}_{\text{trp}}$ in stage 2 reduces the negative impact of incorrectly assigned pseudo-labels. The triplet loss contributes to both improved pseudo-label quality and reduced misclassification errors by training the model to focus on features relevant to ASD, such as changes in machine sounds, while ignoring noise and trivial machine-specific characteristics.

These findings highlight the effectiveness of incorporating triplet loss in all stages of an unsupervised learning framework with pseudo-labels, leading to enhanced model performance.

6 Conclusion

This paper introduced three methods to enhance ASD performance when detailed operational state labels and similar machine data are limited. First, we proposed a pseudo-anomalous set selection method to address scenarios with scarce comparable machine types. By scanning a vast external dataset, we automatically extracted audio samples resembling the target machine's normal sounds, increasing classification complexity without compromising key characteristics. Second, we developed a pseudo-label assignment strategy for unlabeled data, enabling the detection of subtle operational differences critical for ASD. Clustering learned embedding vectors subdivided unlabeled data into pseudo-classes, refining the model's focus on fine-grained anomalies. Third, we implemented iterative learning to refine these techniques, recalculating anomaly scores and improving pseudo-label precision across cycles, progressively boosting detection accuracy.

Experiments were conducted in both unlabeled and labeled settings. In the unlabeled setting, our approach significantly outperformed conventional methods reliant on coarse machine type labels. In the labeled setting, incorporating selected external data further improved detection accuracy. Incorporating external data selected based on similarity to the target's normal sounds consistently enhances detection accuracy, with our targeted selection method proving superior to random selection. For machine types with abundant relevant external data (*large N_{out} machines*), setting the maximum external data volume (N_{max}) to a value comparable to the training data size, such as $N_{\mathrm{max}} = 1000$, optimizes performance, while excessive data leads to degradation. Additionally, employing triplet loss in both training stages improves pseudo-label quality in stage 1 and mitigated the impact of label errors in stage 2, demonstrating its effectiveness in unsupervised learning frameworks.

Collectively, these findings confirmed that our methods robustly improved ASD performance under the limited label availability and for novel machine types, providing practical and effective solutions for industrial applications.

A Appendix

Table A.1: Average AUC (%) of each method under unlabeled (attribute-free) conditions on the DCASE 20222024 Task 2 datasets. source and target denote the two domains; values are mean ± variance over five random seeds.

DCASE	Use label	Method	stage	development		evaluation	
				source	target	source	target
2022	w/ label	Wilkinghoff [60]	1	**86.0±0.9**	78.2±0.7	77.7±0.8	71.6±1.0
		Ba [54]	1	84.9±0.6	78.6±1.7	**80.2±0.6**	**74.2±1.0**
	w/o label	Wilkinghoff [60]	1	69.6±6.1	64.2±5.3	66.9±5.5	63.0±2.2
		Ba [54]	1	71.5±1.2	71.1±1.2	70.4±1.6	66.2±1.4
		Ba+$\mathcal{L}_{\text{trp}}$	1	72.1±1.8	71.7±1.4	70.5±1.6	67.1±0.9
		Ba+$\mathcal{L}_{\text{trp}}$+Ex	2	73.6±1.0	74.9±0.8	71.8±0.5	67.4±1.1
		Ba+$\mathcal{L}_{\text{trp}}$+Ps	2	79.5±1.0	**75.1±0.9**	75.5±0.6	**69.6±1.0**
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	2	79.0±1.1	73.3±1.3	76.4±1.4	68.8±1.2
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	3	80.8±2.0	72.7±2.1	**76.7±0.8**	68.5±2.3
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	4	80.2±2.2	72.8±1.8	76.5±0.4	69.2±0.8
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	5	**82.9±0.6**	74.5±1.8	76.3±0.7	68.9±0.6
2023	w/ label	Wilkinghoff [60]	1	71.2±1.6	75.0±1.5	75.5±0.8	68.7±2.2
		Ba [54]	1	72.0±1.4	74.7±1.5	78.0±1.5	68.3±2.1
	w/o label	Wilkinghoff [60]	1	64.9±1.8	63.6±0.7	62.8±0.8	56.7±1.7
		Ba [54]	1	65.7±1.5	63.5±1.2	60.6±0.9	57.3±1.7
		Ba+$\mathcal{L}_{\text{trp}}$	1	64.6±2.5	64.8±1.4	60.8±1.0	57.8±2.5
		Ba+$\mathcal{L}_{\text{trp}}$+Ex	2	65.7±1.7	64.2±1.1	62.0±1.5	58.9±2.2
		Ba+$\mathcal{L}_{\text{trp}}$+Ps	2	69.5±2.0	67.5±1.2	63.1±1.9	67.9±2.3
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	2	70.0±1.3	67.5±2.5	65.8±1.5	66.2±4.2
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	3	71.1±1.5	**69.3±2.3**	65.9±0.5	70.3±1.7
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	4	71.8±1.2	68.3±1.8	68.6±1.6	**73.3±1.0**
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	5	**72.0±1.7**	68.3±1.1	**68.9±0.8**	72.7±2.0
2024	w/ label	Wilkinghoff [60]	1	68.9±1.3	63.8±1.8	63.2±1.4	62.7±1.7
		Ba [54]	1	74.6±1.1	64.5±1.8	64.0±1.4	65.3±2.4
	w/o label	Wilkinghoff [60]	1	65.9±1.3	58.8±1.4	54.2±1.4	57.4±0.9
		Ba [54]	1	66.1±3.2	59.5±1.8	52.5±1.5	54.6±0.8
		Ba+$\mathcal{L}_{\text{trp}}$	1	65.4±1.9	59.0±1.0	52.6±2.1	55.6±1.6
		Ba+$\mathcal{L}_{\text{trp}}$+Ex	2	68.2±2.4	**61.4±0.9**	54.7±1.1	55.2±0.8
		Ba+$\mathcal{L}_{\text{trp}}$+Ps	2	68.8±1.8	59.6±0.9	57.0±1.8	56.5±1.3
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	2	68.2±1.4	59.8±1.4	58.6±1.1	55.6±0.9
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	3	**71.6±0.0**	58.2±1.1	58.5±1.0	56.2±0.1
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	4	69.9±1.0	59.2±2.7	55.7±0.9	**57.5±0.7**
		Ba+$\mathcal{L}_{\text{trp}}$+Ps+Ex	5	69.9±2.2	59.9±0.4	**59.1±1.4**	56.9±0.7

Table A.2: Average AUC (%) of each method under labeled (attribute-available) conditions on the DCASE 20222024 Task 2 datasets. Notation is identical to Table A.1.

DCASE	Method	stage	development		evaluation	
			source	target	source	target
2022	Wilkinghoff [60]	1	**86.0±0.9**	78.2±0.7	77.7±0.8	71.6±1.0
	Ba [54]	1	84.9±0.6	78.6±1.7	**80.2±0.6**	**74.2±1.0**
	Ba+$\mathcal{L}_{\text{trp}}$	1	83.7±0.6	76.4±1.1	71.2±0.5	66.2±0.5
	Ba+Ex	2	84.7±0.4	**79.4±0.7**	79.9±0.8	73.3±0.8
	Ba+Ps	2	82.9±0.3	76.4±1.4	79.8±0.1	72.4±1.0
	Ba+Ps+$\mathcal{L}_{\text{trp}}$	2	82.7±0.7	77.6±0.8	79.1±0.5	72.4±0.5
	Ba+Ex	3	84.9±0.9	79.0±1.1	79.5±0.7	73.9±0.5
2023	Wilkinghoff [60]	1	71.2±1.6	75.0±1.5	75.5±0.8	68.7±2.2
	Ba [54]	1	72.0±1.4	74.7±1.5	78.0±1.5	68.3±2.1
	Ba+$\mathcal{L}_{\text{trp}}$	1	70.5±2.8	74.3±2.4	71.5±1.7	66.4±1.2
	Ba+Ex	2	**72.1±1.3**	**77.2±0.8**	**79.0±0.3**	**69.2±1.7**
	Ba+Ps	2	71.3±1.0	**77.2±1.4**	75.5±2.4	67.3±2.1
	Ba+Ps+$\mathcal{L}_{\text{trp}}$	2	69.4±0.7	74.9±1.2	70.9±1.9	67.3±2.5
	Ba+Ex	3	70.9±0.8	76.6±1.4	**79.0±1.1**	69.0±1.6
2024	Wilkinghoff [60]	1	68.9±1.3	63.8±1.8	63.2±1.4	62.7±1.7
	Ba [54]	1	74.6±1.1	64.5±1.8	64.0±1.4	65.3±2.4
	Ba+$\mathcal{L}_{\text{trp}}$	1	70.9±1.4	65.4±0.7	60.9±2.9	64.7±1.9
	Ba+Ex	2	**75.2±0.2**	**65.5±1.5**	**64.6±2.2**	**67.3±1.0**
	Ba+Ps	2	72.5±0.7	**65.5±1.9**	60.9±1.1	63.0±2.0
	Ba+Ps+$\mathcal{L}_{\text{trp}}$	2	71.3±1.9	64.9±1.7	59.6±1.5	64.7±1.0
	Ba+Ex	3	74.2±1.5	64.7±1.2	64.5±2.2	67.1±0.8

Biographies

Ibuki Kuroyanagi received his M.E. degree in informatics from Nagoya University, Nagoya, Japan, in 2023. He is currently working toward an Ph.D. degree in informatics at Nagoya University. His research interests include audio signal processing. He is a student member of the Acoustical Society of Japan, and received the Acoustical Society of Japan 2021 Student Presentation Award. In 2022, he received the DCASE Judges' Award.

Takuya Fujimura received his M.E. degree in informatics from Nagoya University, Nagoya, Japan, in 2024. He is currently working toward an Ph.D. degree in informatics at Nagoya University. His research interests include audio signal processing. He is a student member of the Acoustical Society of Japan.

Kazuya Takeda received the B.E. and M.E. degree in electrical engineering and the Dr.Eng. degree from Nagoya University, Nagoya, Japan, in 1983, 1985, and 1994, respectively.,From 1986 to 1989, he was with the Advanced Telecommunication Research (ATR) Laboratories, Osaka, Japan. His research interest at ATR was corpus-based speech synthesis. He was a Visiting Scientist with the Massachusetts Institute of Technology, Cambridge, from November 1987 to April 1988. from 1989 to 1995, he was a Researcher and Research Supervisor with KDD Research and Development Laboratories, Kamifukuoka, Japan. from 1995 to 2003, he was an Associate Professor with the Faculty of Engineering, Nagoya University. Since 2003, he has been a Professor with the Department of Media Science, Graduate School of Information Science, Nagoya University. He is an author or coauthor of more than 100 journal papers, six books, and more than 100 conference proceeding papers. His current research interests are media signal processing and its applications, including spatial audio, robust speech recognition, and driving behavior modeling.,Dr. Takeda was a Conference Technical Cochair of the International Conference on Multimodal Interfaces in 2007 and the International Conference on Vehicular Safety and Electronics in 2009. He was a co-founder of the Biennial Workshop on Digital Signal Processing for In-Vehicle Systems and Safety in 2003.

Tomoki Toda received the B.E. degree from Nagoya University, Nagoya, Japan, in 1999, and the M.E. and D.E. degrees from Nara Institute of Science and Technology (NAIST), Ikoma, Japan, in 2001 and 2003, respectively. He was a Research Fellow with the Japan Society for the Promotion of Science, from 2003 to 2005. He was then an Assistant Professor (20052011) and an Associate Professor (20112015) at NAIST. From 2015, he has been a Professor with the Information Technology Center, Nagoya University. His research interests include statistical approaches to speech processing. He was the recipient of more than 10 paper/achievement awards including the IEEE SPS 2009 Young Author Best Paper Award and the 2013 EURASIP-ISCA Best Paper Award (Speech Communication Journal).

References

[1] 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. Detection and Classification of Acoustic Scenes and Events 2024 Workshop*, 2024, 1–5.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey", *ACM Computing Surveys*, 41(3), 2009, 58 pages, ISSN: 0360-0300.

[3] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges", *IEEE Access*, 6, 2018, 6505–19.

[4] B. Chen, L. Bondi, and S. Das, "Learning to Adapt to Domain Shifts with Few-shot Samples in Anomalous Sound Detection", in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022, 133–9.

[5] H. Chen, L. Ran, X. Sun, and C. Cai, "SW-WAVENET: Learning Representation from Spectrogram and Wavegram Using Wavenet for Anomalous Sound Detection", in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2023, 1–5.

[6] S. Choi and J.-W. Choi, "Noisy-Arcmix: Additive Noisy Angular Margin Loss Combined With Mixup For Anomalous Sound Detection", in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2024, 516–20.

[7] C. Ding, G. Pang, and C. Shen, "Catching Both Gray and Black Swans: Open-set Supervised Anomaly Detection", in *Proc. Computer Vision and Pattern Recognition*, 2022, 7378–88.

[8] K. Dohi, T. Endo, H. Purohit, R. Tanabe, and Y. Kawaguchi, "Flow-Based Self-Supervised Density Estimation for Anomalous Sound Detection", in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2021, 336–40.

[9] 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", in *Proc. Detection and Classification of Acoustic Scenes and Events 2023 Workshop*, 2023, 31–5.

[10] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, and Y. Kawaguchi, "Description and Discussion on DCASE 2022 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques", in *Proc. Detection and Classification of Acoustic Scenes and Events 2022 Workshop*, 5 pages, 2022.

[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. Detection and Classification of Acoustic Scenes and Events 2022 Workshop*, 2022, 1–5.

[12] 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 Ggeneralization Task", in *Proc. Detection and Classification of Acoustic Scenes and Events 2022 Workshop*, 5 pages, 2022.

[13] J. Ebbers, R. Haeb-Umbach, and R. Serizel, “Threshold Independent Evaluation of Sound Event Detection Scores”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2022, 1021–5.

[14] T. Fujimura, K. Imoto, and T. Toda, “Discriminative Neighborhood Smoothing for Generative Anomalous Sound Detection”, in *Proc. European Signal Processing Conference*, 2024, 156–60.

[15] T. Fujimura, I. Kuroyanagi, and T. Toda, “Improvements of Discriminative Feature Space Training for Anomalous Sound Detection in Unlabeled Conditions”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2025, 1–5.

[16] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio Set: An ontology and human-labeled dataset for audio events”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2017, 776–80.

[17] R. Giri, F. Cheng, K. Helwani, S. V. Tenneti, U. Isik, and A. Krishnaswamy, “Group Masked Autoencoder Based Density Estimator for Audio Anomaly Detection”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2020 Workshop*, 2020, 51–5.

[18] J. Guan, Y. Liu, Q. Zhu, T. Zheng, J. Han, and W. Wang, “Time-Weighted Frequency Domain Audio Representation with GMM Estimator for Anomalous Sound Detection”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2023, 1–5.

[19] J. Guan, F. Xiao, Y. Liu, Q. Zhu, and W. Wang, “Anomalous Sound Detection Using Audio Representation with Machine ID Based Contrastive Learning Pretraining”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2023, 1–5.

[20] 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)*, 2023, 191–5.

[21] 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. Detection and Classification of Acoustic Scenes and Events 2023 Workshop*, 2023, 41–5.

[22] 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. Detection and Classification of Acoustic Scenes and Events 2021 Workshop*, 2021, 1–5, ISBN: 978-84-09-36072-7.

[23] T. Hayashi, T. Komatsu, R. Kondo, T. Toda, and K. Takeda, “Anomalous sound event detection based on WavNnet”, in *Proc. European Signal Processing Conference*, 2018, 2494–8.

[24] H. Hojjati and N. Armanfard, “Self-Supervised Acoustic Anomaly Detection Via Contrastive Learning”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2022, 3253–7.

[25] J. Hwang, M. Hira, C. Chen, X. Zhang, Z. Ni, G. Sun, P. Ma, R. Huang, V. Pratap, Y. Zhang, A. Kumar, C.-Y. Yu, C. Zhu, C. Liu, J. Kahn, M. Ravanelli, P. Sun, S. Watanabe, Y. Shi, and Y. Tao, “TorchAudio 2.1: Advancing Speech Recognition, Self-Supervised Learning, and Audio Processing Components for Pytorch”, in *Proc. Automatic Speech Recognition and Understanding*, 2023, 1–9.

[26] A. Jiang, W.-Q. Zhang, Y. Deng, P. Fan, and J. Liu, “Unsupervised Anomaly Detection and Localization of Machine Audio: A Gan-Based Approach”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2023, 1–5.

[27] S. Kapka, “ID-Conditioned Auto-Encoder for Unsupervised Anomaly Detection”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2020 Workshop*, 2020, 71–5.

[28] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, “Description and Discussion on DCASE 2021 Challenge Task 2: Unsupervised Anomalous Detection for Machine Condition Monitoring Under Domain Shifted Conditions”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2021 Workshop*, 2021, 186–90.

[29] Y. Kawaguchi, R. Tanabe, T. Endo, K. Ichige, and K. Hamada, “Anomaly Detection Based on an Ensemble of Dereverberation and Anomalous Sound Extraction”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2019, 865–9.

[30] P. Kirichenko, P. Izmailov, and A. G. Wilson, “Why Normalizing Flows Fail to Detect Out-of-Distribution Data”, in *Proc. International Conference on Neural Information Processing Systems*, 2020, 20578–89.

[31] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, “Description and Discussion on DCASE2020 Challenge Task2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2020 Workshop*, 2020, 81–5.

[32] 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. Workshop on Applications of Signal Processing to Audio and Acoustics*, 2019, 308–12.

[33] I. Kuroyanagi, T. Hayashi, Y. Adachi, T. Yoshimura, K. Takeda, and T. Toda, “An Ensemble Approach to Anomalous Sound Detection Based on Conformer-Based Autoencoder and Binary Classifier Incorporated with Metric Learning”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2021 Workshop*, 2021, 110–4.

[34] I. Kuroyanagi, T. Hayashi, K. Takeda, and T. Toda, “Anomalous Sound Detection Using a Binary Classification Model and Class Centroids”, in *Proc. European Signal Processing Conference*, 2021, 1995–9.

[35] I. Kuroyanagi, T. Hayashi, K. Takeda, and T. Toda, “Two-stage anomalous sound detection systems using domain generalization and specialization techniques”, *tech. rep.*, 5 pages, DCASE2022 Challenge, 2022.

[36] I. Kuroyanagi and T. Komatsu, “Self-Supervised Learning Method Using Multiple Sampling Strategies for General-Purpose Audio Representation”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2022, 3263–7.

[37] W. Liu, D. Cui, Z. Peng, and J. Zhong, “Outlier Detection Algorithm Based on Gaussian Mixture Model”, in *Proc. International Conference on Power, Intelligent Computing and System*, 2019, 488–92.

[38] Y. Liu, J. Guan, Q. Zhu, and W. Wang, “Anomalous Sound Detection Using Spectral-Temporal Information Fusion”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2022, 816–20.

[39] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization”, in *Proc. International Conference on Learning Representations*, 8 pages, 2019.

[40] 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”, *tech. rep.*, 4 pages, DCASE2024 Challenge, 2024.

[41] D. K. McClish, “Analyzing a Portion of the ROC Curve”, *Medical Decision Making*, 9(3), 1989, 190–5.

[42] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, “VT-ADL: A Vision Transformer Network for Image Anomaly Detection and Localization”, in *Proc. International Symposium on Industrial Electronics*, 2021, 1–6.

[43] 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. Detection and Classification of Acoustic Scenes and Events 2024 Workshop*, 2024, 106–10.

[44] T. Nishida, K. Dohi, T. Endo, M. Yamamoto, and Y. Kawaguchi, “Anomalous Sound Detection Based on Machine Activity Detection”, in *Proc. European Signal Processing Conference*, 2022, 269–73.

[45] 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 2024 Challenge Task 2: First-Shot Unsupervised Anomalous Sound Detection for Machine Condition Monitoring”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2024 Workshop*, 2024, 111–5.

[46] G. Papamakarios, T. Pavlakou, and I. Murray, “Masked Autoregressive Flow for Density Estimation”, in *Proc. International Conference on Neural Information Processing Systems*, 2017, 2335–44, ISBN: 9781510860964.

[47] 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. Detection and Classification of Acoustic Scenes and Events 2020 Workshop*, 2020, 170–4.

[48] H. Purohit, R. Tanabe, T. Endo, K. Suefusa, Y. Nikaido, and Y. Kawaguchi, “Deep Autoencoding GMM-Based Unsupervised Anomaly Detection in Acoustic Signals and its Hyper-Parameter Optimization”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2020 Workshop*, 2020, 175–9.

[49] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, “MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2019 Workshop*, 2019, 209–13.

[50] D. Reynolds, “Gaussian Mixture Models”, in, *Encyclopedia of Biometrics*, 2009, 659–63, ISBN: 978-0-387-73003-5.

[51] L. Ruff, R. A. Vandermeulen, B. J. Franks, K.-R. Müller, and M. Kloft, “Rethinking Assumptions in Deep Anomaly Detection”, in *ICML 2021 Workshop on Uncertainty & Robustness in Deep Learning*, 2021.

[52] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, “Deep Semi-Supervised Anomaly Detection”, in *Proc. International Conference on Learning Representations*, 23 pages, 2020.

[53] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, “Anomalous Sound Detection Based on Interpolation Deep Neural Network”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2020, 271–5.

[54] F. Takuya, I. Kuroyanagi, and T. Toda, “The NU systems for DCASE 2024 Challenge Task 2”, *tech. rep.*, 5 pages, DCASE2024 Challenge, 2024.

[55] R. Tanabe, H. Purohit, K. Dohi, T. Endo, Y. Nikaido, T. Nakamura, and Y. Kawaguchi, “MIMII DUE: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection with Domain Shifts due to Changes in Operational and Environmental Conditions”, *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021, 21–5.

[56] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. S. Yu, “Generalizing to Unseen Domains: A Survey on Domain Generalization”, *IEEE Transactions on Knowledge and Data Engineering*, 35(8), 2023, 8052–72.

[57] Y. Wang, X. Deng, J. Jiang, and Q. Kong, “ANOMALOUS SOUND DETECTION BASED ON PSEUDO LABELS FROM GUIDED CLUSTERING”, *tech. rep.*, 3 pages, DCASE2024 Challenge, 2024.

[58] K. Wilkinghoff, “Combining Multiple Distributions based on Sub-Cluster AdaCos for Anomalous Sound Detection under Domain Shifted Conditions”, in *Proc. Detection and Classification of Acoustic Scenes and Events 2021 Workshop*, 2021, 55–9, ISBN: 978-84-09-36072-7.

[59] K. Wilkinghoff, “Design Choices for Learning Embeddings from Auxiliary Tasks for Domain Generalization in Anomalous Sound Detection”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2023, 1–5.

[60] K. Wilkinghoff, “Self-Supervised Learning for Anomalous Sound Detection”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2024, 276–80.

[61] K. Wilkinghoff and F. Fritz, “On Using Pre-Trained Embeddings for Detecting Anomalous Sounds with Limited Training Data”, in *Proc. European Signal Processing Conference*, 2023, 186–90.

[62] K. Wilkinghoff and F. Kurth, “Why Do Angular Margin Losses Work Well for Semi-Supervised Anomalous Sound Detection?”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32, 2024, 608–22, ISSN: 2329-9304.

[63] X. Xia, X. Pan, N. Li, X. He, L. Ma, X. Zhang, and N. Ding, “GAN-Based Anomaly Detection: A Review”, *Neurocomputing*, 493(C), 2022, 497–535, ISSN: 0925-2312.

[64] Y.-Y. Yang, M. Hira, Z. Ni, A. Astafurov, C. Chen, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Hwang, J. Chen, P. Goldsborough, S. Narenthiran, S. Watanabe, S. Chintala, and V. Quenneville-Bélair, “Torchaudio: Building Blocks for Audio and Speech Processing”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2022, 6982–6.

[65] X. Yao, R. Li, J. Zhang, J. Sun, and C. Zhang, “Explicit Boundary Guided Semi-Push-Pull Contrastive Learning for Supervised Anomaly Detection”, in *Proc. Computer Vision and Pattern Recognition*, 2023, 24490–9.

[66] X.-M. Zeng, Y. Song, Z. Zhuo, Y. Zhou, Y.-H. Li, H. Xue, L.-R. Dai, and I. McLoughlin, “Joint Generative-Contrastive Representation Learning for Anomalous Sound Detection”, in *Proc. International Conference on Acoustics, Speech and Signal Processing*, 2023, 1–5.

[67] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond Empirical Risk Minimization”, in *Proc. International Conference on Learning Representations*, 13 pages, 2018.