



# EFFICIENT STATE-SPACE MODEL FOR AUDIO ANOMALY DETECTION WITH DOMAIN ADAPTATION

Taharim Rahman Anon<sup>1</sup>, Jakaria Islam Emon<sup>1</sup>

<sup>1</sup>Hokkaido Denshikiki Co., Ltd., Sapporo, Hokkaido, Japan  
tahrim.anon21@gmail.com, emon\_j@hdkcs.co.jp

**Abstract**—This paper presents a two-stage, embedding-centric framework for Unsupervised Anomaly Sound Detection (UASD), specifically addressing the challenges of first-shot generalization and computational efficiency. Our approach utilizes an efficient state-space model (SSM) backbone. Pre-training of this backbone is accelerated using a pixel-unshuffle (space-to-depth) input transformation for spectrograms, which reduces training time by approximately 87.5% while preserving representation quality. Subsequently, the pre-trained model is fine-tuned with a specialized anomaly head that fuses multi-level features, combined with pseudo outlier exposure and domain-adversarial adaptation employing a gradient reversal layer. Our system demonstrates superior performance over the DCASE 2025 autoencoder baseline. Machine-specific models achieve a harmonic mean total score of 0.722. This work establishes the efficacy of SSMs for this task and offers a scalable, robust solution for UASD in dynamic acoustic environments.

**Index Terms**—audio anomaly detection, pseudo outlier-exposure, domain-adversarial training, State Space Model, Representation learning

## 1. INTRODUCTION

The *Detection and Classification of Acoustic Scenes and Events* (DCASE) Challenge Task 2 in particular has steadily raised the bar for unsupervised anomaly sound detection (UASD). The task evolved from plain UASD in 2020 [1], [2], through domain adaptation in 2021 [3], [4], [5], domain generalization in 2022 [6], and, most recently, the demanding *first-shot* scenario in 2023–2025 [7], [8]. Current systems typically adopt either *inlier modelling* with autoencoders (AE) [9]–[11] or *outlier exposure* (OE) [12], [13], where auxiliary or pseudo-outlier data improve robustness [14], [15].

Most existing backbones CNNs [16], [17], [18], AEs [6], diffusion models [19], and lightweight nets such as MobileFaceNet [20], [21] struggle to model long-range temporal context. Transformer variants mitigate that with self-attention [22], [23], but for sequence length  $N$ , their quadratic time–memory cost  $O(N^2)$  limits practical use on long audio streams. Modern state-space models (SSMs) offer a compelling alternative: they scale linearly,  $O(N)$ , while retaining strong sequence modelling power [24]. Yet, a recent survey of Task 2 work reveals no SSM backbones to date, leaving a clear gap.

To bridge this gap, we introduce a UASD system tailored to the first-shot, domain-generalisation setting of DCASE Task 2 [10]. Our method couples an efficient SSM backbone with (i) a space-to-depth [25] spectrogram rearrangement that accelerates pre-training, and (ii) a fine-tuning stage that blends pseudo-outlier exposure with gradient-reversal-based domain adaptation [26]. We investigate both *machine-specific* and *machine-generalised* variants during fine-tuning.

**Our contributions are:**

- 1) **First SSM backbone for DCASE Task 2.** We present the first efficient state-space model applied to the DCASE first-shot UASD challenge.
- 2) **Faster pre-training via space-to-depth.** A novel spectrogram rearrangement cuts pre-training time by about 87.5 % without degrading representation quality.

- 3) **Compact anomaly head with domain adaptation.** We design a lightweight head that fuses multi-level features and pair it with pseudo OE plus a gradient-reversal layer for domain alignment.

Extensive experiments confirm that our system surpasses the official DCASE 2025 baseline [8].

**Paper structure:** Section 2 details the architecture and training procedure; Section 3 describes datasets and metrics; Section 4 reports results and ablations; and Section 5 summarises findings and future directions.

## 2. METHOD

Our method tackles audio anomaly detection using a two-stage representation learning strategy:

**Stage I: Pre-training.** The first stage focuses on learning robust general-purpose acoustic features. We pre-train an efficient bidirectional Audio-Mamba [27] encoder using a supervised classification objective.

**Stage II: Fine-tuning with pOE and Domain Adaptation.** The pre-trained encoder is repurposed for anomaly detection in two steps. First, we swap the classification head for an anomaly head that fuses global and intermediate features. We then fine-tune the network with a pseudo Outlier Exposure (pOE) loss: embeddings of normal samples are pulled toward a learned centre, while embeddings of auxiliary pseudo-outliers are pushed away (Section 2.5). In parallel, domain-adversarial training encourages domain-invariant representations, improving robustness across operating conditions.

### 2.1. Spectrogram Tokenization via Space-to-Depth

Let  $X \in \mathbb{R}^{C \times H \times W}$  be a log-mel spectrogram ( $C=1$ ). We first apply *pixel-unshuffle* with factor  $r$  to fold local time–frequency context into the channel dimension:

$$X' = \text{PU}(X, r) \in \mathbb{R}^{r^2 C \times \frac{H}{r} \times \frac{W}{r}}. \quad (1)$$

With  $r=4$  the  $128 \times 1024$  input becomes  $16 \times 32 \times 256$ , reducing the token length by  $r^2$  while preserving locality inside the enlarged channel dimension, as illustrated in Fig. 1.

We then partition  $X'$  into non-overlapping patches of size  $p \times p$  (with  $p=16$ ), flatten each patch  $S_i \in \mathbb{R}^{p^2 r^2 C}$  using  $\text{vec}(\cdot)$  and project it linearly into a  $D$ -dimensional embedding,

$$E_i = W_{\text{patch}} \text{vec}(S_i) + \mathbf{b}_{\text{patch}}, \quad (2)$$

yielding the token sequence  $E = [E_1, \dots, E_N] \in \mathbb{R}^{N \times D}$  with  $N = \frac{HW}{p^2 r^2}$ .

### 2.2. Bidirectional State–Space Encoder

The patch-embedded token sequence  $E = [E_1, \dots, E_N] \in \mathbb{R}^{N \times D}$  is processed by a stack of  $K$  identical Forward-Bidirectional Audio Mamba (AuM) blocks, as depicted in Fig. 2. Each AuM block executes a sequence of operations to transform its input  $x_t$ .

First, an input projection maps  $x_t$  to an intermediate representation  $\tilde{x}_t = W_{\text{in}} x_t + b_{\text{in}}$ . This projected sequence  $\tilde{X}$  is then fed into a

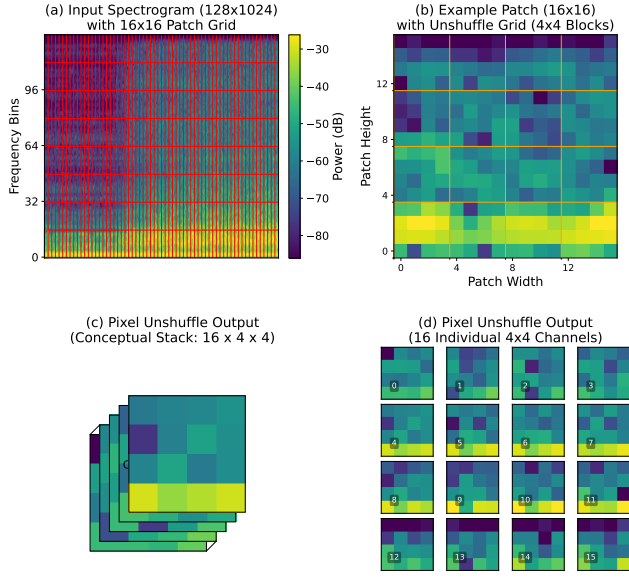


Fig. 1: Pixel unshuffle ( $r = 4$ ) applied to spectrogram patches. (a) Input spectrogram ( $128 \times 1024$ ) with  $16 \times 16$  patch grid. (b)  $16 \times 16$  input patch with  $4 \times 4$  unshuffle blocks. (c) Conceptual output stack ( $16 \times 4 \times 4$ ). (d) The 16 resulting  $4 \times 4$  feature-map channels. The operation converts spatial resolution ( $H, W$ ) to channel depth ( $r^2 C$ ), reducing the number of tokens per patch from  $16 \times 16 = 256$  to  $4 \times 4 = 16$  (a factor of  $r^2 = 16$ ).

depth-wise Conv1D generator. This convolutional layer produces the time-variant parameters for the core State-Space Model (SSM) along with a pre-activation gate,  $(\bar{A}_t, \bar{B}_t, \Delta_t, \tilde{g}_t) = \text{Conv1D}(\tilde{X})_t$ . Here,  $\bar{A}_t$  is the time-varying discrete state-transition matrix (state  $\rightarrow$  state),  $\bar{B}_t$  is the discrete input matrix (input  $\rightarrow$  state),  $\Delta_t > 0$  is the learnable discretization step used to form  $\bar{A}_t$  and  $\bar{B}_t$ , and  $\tilde{g}_t$  is the pre-activation of the fusion gate (with  $g_t = \sigma(\tilde{g}_t)$ ). The actual gate  $g_t = \sigma(\tilde{g}_t) \in (0, 1)^{D_s}$  is obtained via a sigmoid activation, where  $D_s$  denotes the SSM state size.

The central component of the block is a Bidirectional SSM scan, which operates in linear time. Using the generated parameters  $\bar{A}_t$  and  $\bar{B}_t$ , and a shared output projection matrix  $C$ , the forward and backward hidden states ( $h_t^f, h_t^b$ ) and their corresponding outputs ( $y_t^f, y_t^b$ ) are computed. The output is derived from the current (updated) state:

$$h_t^f = \bar{A}_t h_{t-1}^f + \bar{B}_t \tilde{x}_t, \quad y_t^f = C h_t^f, \quad (3a)$$

$$h_t^b = \bar{A}_t h_{t+1}^b + \bar{B}_t \tilde{x}_t, \quad y_t^b = C h_t^b. \quad (3b)$$

These directional outputs are subsequently combined through element-wise gated fusion:  $y_t = g_t \odot y_t^f + (1 - g_t) \odot y_t^b$ . Finally, an output projection and residual connection yield the block's output:  $z_t = W_{\text{out}} y_t + b_{\text{out}}$ , leading to  $X_t^{\text{next}} = x_t + z_t$ . This entire structure is followed by a layer normalization step and an MLP sub-block.

The discrete-time SSM utilized in (3) is conceptually derived from an underlying continuous-time formulation,  $\frac{dh(t)}{dt} = Ah(t) + Bx(t)$ , with  $y(t) = Ch(t)$ . This continuous system is discretized using the learnable step  $\Delta_t$  (which is one of the parameters generated by the Conv1D layer). The resulting discrete-time transition matrices  $\bar{A}_t$  and input matrices  $\bar{B}_t$  are formulated as  $\bar{A}_t = I + \Delta_t A_t$  and  $\bar{B}_t = \Delta_t B_t$ . This bidirectional SSM architecture efficiently captures global context

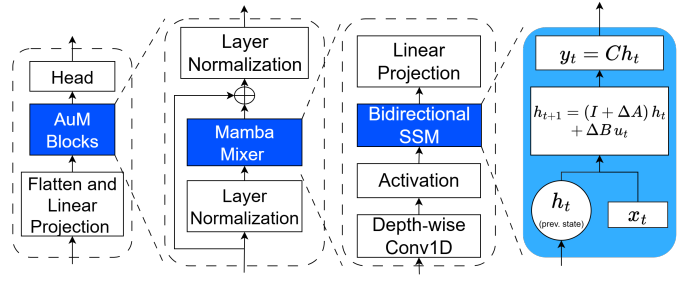


Fig. 2: Four-level view of the Bidirectional AuM encoder: (A) Encoder stack; (B) one AuM block; (C) internal Mamba mixer; (D) state-space cell used by the bidirectional selective scan.

with an overall computational complexity of  $\mathcal{O}(ND_s)$  in both time and memory.

### 2.3. Design of Anomaly Head

The anomaly head is designed to combine global and intermediate encoder features to produce a compact representation used for anomaly scoring. Let  $h^{(0)} \in \mathbb{R}^d$  be the global feature vector from the final encoder layer, and let  $\{h^{(m)}\}_{m=1}^M$  be pooled intermediate features from  $M$  selected layers. A weighted fusion of intermediate features is computed as:

$$\hat{h} = \sum_{m=1}^M \beta_m h^{(m)}, \quad \beta = \text{softmax}(\alpha), \quad (4)$$

with  $\alpha \in \mathbb{R}^M$  as learnable fusion weights. The fused representation is then computed as the  $\ell_2$ -normalized sum:

$$\tilde{h} = \frac{h^{(0)} + \hat{h}}{\|h^{(0)} + \hat{h}\|_2}. \quad (5)$$

This fused feature  $\tilde{h}$  is passed through a two-layer MLP with batch normalization, ReLU activation, and dropout:

$$z = W_2 \text{ReLU}(\text{BN}(\text{Dropout}_p(W_1 \tilde{h} + b_1))) + b_2, \quad (6)$$

where  $W_1, W_2$  and  $b_1, b_2$  are learnable parameters. The output  $z \in \mathbb{R}^D$  is the final anomaly embedding used in downstream scoring.

### 2.4. Stage 1: Supervised Pre-training

The initial stage focuses on learning robust and general-purpose acoustic representations. The Bidirectional State-Space Encoder, appended with a linear classification head, is pre-trained on all available normal clips using data from the DCASE 2022-2025 Task-2 datasets (as detailed in Section 3.1). The optimization proceeds as follows:

- Input spectrograms are first processed using the pixel-unshuffle operation (as detailed in (1)) prior to patch embedding. This step reduces the sequence length of tokens fed to the encoder.
- The model, comprising the backbone encoder and the classification head, is optimized to minimize the standard cross-entropy loss:

$$L_{\text{cls}} = -\frac{1}{B} \sum_{i=1}^B \sum_{j=1}^{C_{\text{cls}}} y_{ij} \log p(y_{ij} | x_i), \quad (7)$$

where  $B$  is the batch size,  $C_{\text{cls}}$  is the number of classes (i.e., machine types) in the pre-training dataset,  $y_{ij}$  is a binary indicator if sample  $i$  belongs to class  $j$ , and  $p(y_{ij} | x_i)$  is the predicted softmax probability of class  $j$  for input  $x_i$ .

## 2.5. Stage 2: Domain-Adaptive Fine-Tuning

In the second stage, we adapt our pre-trained backbone for anomaly detection under domain shift. The majority of the backbone layers are frozen, with only the final  $K$  blocks being unfrozen for fine-tuning.

Two primary components are introduced for this stage: the anomaly head  $g$ , which processes features from the backbone to produce the final embedding  $\hat{z} \in \mathbb{R}^D$ , and a domain classifier  $D$ , which takes  $\hat{z}$  as input to predict its domain label. The fine-tuning objective combines an anomaly detection loss, leveraging pseudo Outlier Exposure (pOE), with an adversarial domain adaptation loss.

**Anomaly Objective (pOE with SVDD-inspired Center) [28]:** To anchor the representation of normal data, an estimate of the normal data cluster center,  $c \in \mathbb{R}^D$ , is maintained. This center is updated using momentum  $\mu$  with embeddings from normal samples:

$$c \leftarrow (1 - \mu) \cdot c + \mu \cdot \hat{z}_{\text{normal}}, \quad (8)$$

where  $\hat{z}_{\text{normal}}$  is the embedding  $\hat{z}$  derived from a normal input sample. The anomaly loss  $\mathcal{L}_{\text{oe}}$  is formulated to attract embeddings of normal samples ( $\hat{z}_{y=0}$ ) towards this center  $c$ , while simultaneously repelling embeddings of pOE samples ( $\hat{z}_{y=1}$ ) from it, based on cosine similarity:

$$\mathcal{L}_{\text{oe}} = \mathbb{E}_{\hat{z}_{y=0}} [1 - \cos(\hat{z}_{y=0}, c)] + \mathbb{E}_{\hat{z}_{y=1}} [\max(0, \cos(\hat{z}_{y=1}, c) - \delta)], \quad (9)$$

where  $y = 0$  denotes normal samples and  $y = 1$  denotes pOE samples.

**Domain Adaptation Objective (Adversarial Loss):** To encourage the model to learn domain-invariant representations, the embedding  $\hat{z}$  is passed through a Gradient Reversal Layer (GRL) before it serves as input to the domain classifier  $D$ . The adversarial domain classification loss  $\mathcal{L}_{\text{adv}}$  is then defined using the standard cross-entropy loss  $\mathcal{L}_{\text{ce}}$ :

$$\mathcal{L}_{\text{adv}} = \mathcal{L}_{\text{ce}}(D(\text{GRL}(\hat{z})), d), \quad (10)$$

where  $d$  represents the true domain label of the input sample. During backpropagation, the GRL inverts the sign of the gradients flowing back to the feature extractor as illustrated in Fig. 3. This trains the feature extractor (comprising the unfrozen backbone layers and the anomaly head  $g$ ) to generate embeddings  $\hat{z}$  that hinder the domain classifier's ability to distinguish between domains.

**Combined Objective and Optimization Strategy:** The parameters of the unfrozen final  $K$  backbone blocks (denoted  $\theta_{b_K}$ ) and the anomaly head ( $\theta_g$ ) collectively referred to as the feature extractor parameters  $\theta_f = \{\theta_{b_K}, \theta_g\}$  are updated by minimizing the combined loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{oe}} + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}, \quad (11)$$

where  $\lambda_{\text{adv}}$  is a hyperparameter balancing the contribution of the adversarial domain loss.

A two-optimizer approach is employed for stable adversarial training:

- The first optimizer updates the feature extractor parameters  $\theta_f$  by minimizing  $\mathcal{L}_{\text{total}}$ .
- The second optimizer updates the parameters of the domain classifier  $D$  (denoted  $\theta_D$ ) by minimizing its classification loss  $\mathcal{L}_{\text{ce}}(D(\text{detach}(\hat{z})), d)$ . The `detach()` operation ensures that gradients from this optimization step do not propagate back to the feature extractor  $\theta_f$ .

## 3. EXPERIMENTAL SETUP

### 3.1. Data

All experiments utilized audio data from the DCASE 2022-2025 Task 2 corpus [1], [2], [29], focusing on the seven core machine

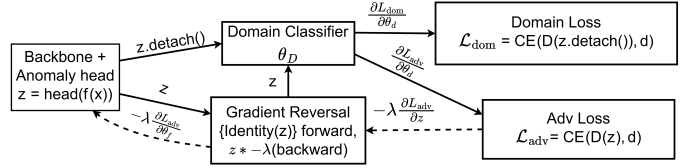


Fig. 3: Architecture of the domain-adversarial branch. Embedding  $\hat{z}$  from the feature extractor ( $\theta_f$ ) is classified by domain  $D$  ( $\theta_D$ ).

types: *ToyCar*, *ToyTrain*, *Fan*, *Gearbox*, *Bearing*, *Slider*, *Valve*. For pre-training, we constructed a dataset using all available **normal clips** for these seven machine types from the 2022-2025 development splits, no test-set audio was used. This resulted in approximately 62,500 clips for training the pre-trained models. A separate set of 2,800 clips from the same development splits, also consisting of normal data from these core machine types, was reserved for the validation of the pre-trained models. Subsequently, for fine-tuning, normal data was prepared in two distinct configurations. For *machine-specific* models, dedicated normal datasets were curated for each of the seven machine types. Each such dataset comprised 1,000 normal clips, which were domain-balanced by oversampling target domain samples to match source domain counts. For the *machine-generalized* model, a single, larger dataset of normal clips was formed. This dataset was created by pooling together individually domain-balanced normal clips prepared from all seven core machine types and all additional data provided. Finally, model performance was evaluated using the official DCASE 2025 Task 2 test sets.

### 3.2. Implementation Details

All audio signals were resampled to 16 kHz. We extracted 128-bin log-Mel spectrograms using a 1024-sample frame length and using a hop length tuned to yield 1024 frames for a 10-second audio clip. The pixel-unshuffle operation (detailed in Section 2.1, (1)) was applied with a factor  $r = 4$  to the spectrograms prior to patch embedding. All models were developed using PyTorch and trained on a single NVIDIA RTX 3090 GPU.

**Stage 1: Pre-training:** The backbone encoder, attached with a linear classification head, was pre-trained for 5 epochs to classify machine types based on their normal operational sounds. For this stage, we employed the AdamW optimizer with a learning rate of  $1 \times 10^{-4}$ , a weight decay of  $1 \times 10^{-2}$ , and a batch size of 256.

**Stage 2: Fine-tuning:** In this stage, the AdamW optimizer was used for updating both the feature extractor parameters ( $\theta_f = \{\theta_{b_K}, \theta_g\}$ ) and the separate domain classifier parameters ( $\theta_D$ ). Key hyperparameters for training the feature extractor (such as learning rates, regularization settings, loss component weights e.g.,  $\lambda_{\text{adv}}$ , and the number of unfrozen layers  $K$ ) were determined using BOHB (Bayesian Optimization with Hyperband), which couples a model-based sampler with Hyperband's early-stopping schedule; we maximized validation AUC under an epoch-based budget using a Hyperband reduction factor of  $\eta = 3$  [30]. For the machine-specific models, hyperparameters were tuned separately for each machine type; the machine-generalized model underwent a separate search. For the domain classifier optimizer, we used a fixed learning rate of  $1 \times 10^{-4}$  and a weight decay of  $1 \times 10^{-2}$ . During fine-tuning, a batch size of 256 was used with gradient accumulation over 2 steps. Gradients for  $\theta_f$  were clipped at a maximum L2 norm of 1.0. The momentum  $\mu$  for updating the SVDD-inspired center  $c$  (used in  $\mathcal{L}_{\text{oe}}$ ) was set to 0.99. Models were trained for a maximum of 100 epochs, with early stopping initiated if the validation AUC showed no improvement for 15 consecutive epochs. The hyperparameters found through the

Table 1: Ablation studies on key framework components.

Component	Configuration	Time/Latency	AUC(src)	AUC(tgt)	pAUC
Pixel Unshuffle	$r = 4$ (Proposed)	90 min	–	–	–
	$r = 1$ (No Unshuffle)	720 min	–	–	–
Backbone	Audio-Mamba	22.7ms	0.89	0.84	0.66
	AST Baseline	26.1ms	0.83	0.85	0.68
Pre-training	With (Proposed)	–	0.89	0.84	0.60
	Without	–	0.63	0.50	0.52
Fine-tuning	pOE + GRL (Full)	–	<b>0.89</b>	<b>0.84</b>	<b>0.66</b>
	pOE only	–	0.87	0.79	0.63
	OC-SVDD + GRL	–	0.80	0.81	0.61
	OC-SVDD only	–	0.66	0.78	0.58

optimization process were used for the final training and evaluation of the fine-tuned models.

### 3.3. Evaluation Metrics

To evaluate anomaly detection performance, we adopt two standard metrics: the Area Under the Receiver Operating Characteristic Curve (AUC) and the partial AUC (pAUC). Both metrics are reported for source and target domains to assess generalization under domain shift. To aggregate performance across the seven machine classes, we compute the harmonic mean (hmean) and arithmetic mean (amean) of the per-class scores.

## 4. RESULTS AND DISCUSSION

### 4.1. Ablation Studies

Table 1 confirms that each design choice contributes either speed, accuracy, or both.

**Input rearrangement.** Pixel-unshuffle with scale  $r=4$  cuts end-to-end pre-training time from 720 to 90 min (−87.5 %) while the mean average precision falls by less than 0.5 pp (0.9934 → 0.9891). This trade-off is attractive for industrial pipelines that need frequent model updates.

**Backbone.** Our Audio-Mamba encoder yields a higher source-domain AUC than the AST baseline (0.89 vs 0.83) [31] and, crucially, runs 13 % faster at inference time (22.7 ms vs 26.1 ms). Although AST scores marginally better on target-domain AUC, Audio-Mamba offers a better overall hmean owing to its stronger pAUC.

**Pre-training.** Removing the contrastive pre-training stage drops target-domain AUC from 0.84 to 0.50 and pAUC from 0.60 to 0.52, highlighting that representation learning is critical when only one normal recording per machine is available.

**Fine-tuning strategy.** The full pOE + GRL recipe attains the best target AUC (0.84). GRL alone raises domain robustness by around 0.05 AUC, while pOE outperforms OC-SVDD regardless of whether GRL is present, confirming the value of synthetic outlier generation.

### 4.2. Performance Comparison

Table 2 summarises end-to-end accuracy. Due to limited computational resources, all results presented were obtained from a single experimental run, precluding the calculation of statistical significance intervals. Our *machine-specific* system lifts the harmonic-mean TOTAL score from 0.641 to 0.722, a 12.6 % relative gain, and the arithmetic mean from 0.660 to 0.740. Source-domain AUC rises from 0.681 to 0.812, target-domain AUC from 0.614 to 0.702, and pAUC from 0.628 to 0.651. Per-machine results in Table 3 show that the largest absolute jump occurs on *Fan*, where source-domain AUC climbs from 0.644 to 0.980 and target-domain AUC from 0.338 to 0.880. These gains indicate that the combination of an SSM backbone and pOE is particularly effective on highly non-stationary signals.

The *machine-generalised* model reaches a TOTAL of 0.600 about 6.4 % below the baseline. Training one network on all seven machine

Table 2: Performance comparison on seven core machine types. TOTAL score is arithmetic mean of hmean values across three metrics.

System	Metric	hmean	amean
<b>Proposed</b> (machine-specific)	AUC (source)	0.812	0.832
	AUC (target)	0.702	0.725
	pAUC	0.651	0.663
	<b>TOTAL</b>	<b>0.722</b>	<b>0.740</b>
<b>Proposed</b> (machine-generalized)	AUC (source)	0.643	0.666
	AUC (target)	0.603	0.633
	pAUC	0.552	0.563
	<b>TOTAL</b>	<b>0.600</b>	<b>0.621</b>
<b>Baseline</b> (AE)	AUC (source)	0.681	0.702
	AUC (target)	0.614	0.631
	pAUC	0.628	0.646
	<b>TOTAL</b>	<b>0.641</b>	<b>0.660</b>

Table 3: Detailed per-machine AUC results for source and target domains.

System	Domain	ToyCar	ToyTrain	Fan	Gearbox	Bearing	Slider	Valve
Proposed (specific)	Source	0.894	0.736	0.980	0.958	0.728	0.900	0.626
	Target	0.848	0.620	0.880	0.770	0.794	0.640	0.521
Proposed (generalized)	Source	0.820	0.870	0.580	0.610	0.710	0.500	0.570
	Target	0.810	0.870	0.500	0.490	0.570	0.680	0.510
Baseline (AE)	Source	0.790	0.673	0.644	0.702	0.711	0.703	0.650
	Target	0.725	0.598	0.338	0.653	0.600	0.575	0.611

types dilutes machine-specific cues and stresses the model capacity. Even so, it exceeds the baseline on *ToyTrain* and cuts the domain gap on *Slider* (source 0.500, target 0.680), suggesting that a unified model could become competitive if equipped with larger hidden widths or domain-aware loss weighting.

Overall, these results confirm that (i) state-space backbones scale favourably on long audio, (ii) pOE + GRL is an effective fine-tuning strategy for first-shot UASD, and (iii) machine-specific deployment currently offers the best accuracy-latency trade-off for on-device condition monitoring.

## 5. CONCLUSION AND FUTURE WORK

We introduced a two-stage embedding-centric framework for Unsupervised Anomaly Sound Detection. The proposed system integrates an efficient AudioMamba State Space Model backbone, accelerated pre-training through space-to-depth spectrogram tokenization (pixel-unshuffle), and a robust domain-adaptive fine-tuning strategy. This fine-tuning stage effectively combines a compact anomaly head that fuses multi-level features with pseudo Outlier Exposure and GRL based domain adaptation techniques. Our comprehensive experiments and ablation studies validated the efficacy of each component. We demonstrated this to be the first application of an SSM backbone for the DCASE Task 2 first-shot UASD challenge, achieving significant improvements in AUC and pAUC metrics over established baselines. The substantial reduction in pre-training time due to pixel-unshuffle, and the enhanced robustness and domain generalization capabilities imparted by the pOE and GRL techniques during fine-tuning, were clearly evidenced.

Several directions look promising. First, experimenting with larger or alternative state-space architectures may unlock further accuracy and speed gains. Second, we will refine pseudo-outlier generation in pOE so it spans a wider range of real-world anomalies. Third, introducing domain-aware loss weighting could make a single, unified model competitive with machine-specific ones. Finally, a fully self-supervised pre-training stage for the backbone would eliminate the need for labelled machine types, easing deployment where annotated data are scarce.

## 6. ACKNOWLEDGMENT

This work was supported by Hokkaido Denshikiki Co., Ltd.

## REFERENCES

- [1] 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 *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, November 2019, pp. 209–213. [Online]. Available: [http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop\\_Purohit\\_21.pdf](http://dcase.community/documents/workshop2019/proceedings/DCASE2019Workshop_Purohit_21.pdf)
- [2] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, “ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection,” in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, November 2019, pp. 308–312. [Online]. Available: <https://ieeexplore.ieee.org/document/8937164>
- [3] 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 *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 186–190.
- [4] 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)*, pp. 21–25, 2021.
- [5] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, “ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions,” in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 1–5.
- [6] 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 *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022, pp. 1–5.
- [7] T. Nishida, N. Harada, D. Niizumi, D. Albertini, R. Sannino, S. Pradolini, F. Augusti, K. Imoto, K. Dohi, H. Purohit, T. Endo, and Y. Kawaguchi, “Description and discussion on DCASE 2025 challenge task 2: First-shot unsupervised anomalous sound detection for machine condition monitoring,” *In arXiv e-prints: 2506.10097*, 2025.
- [8] N. Harada, D. Niizumi, Y. Ohishi, D. Takeuchi, and M. Yasuda, “First-shot anomaly sound detection for machine condition monitoring: A domain generalization baseline,” in *2023 31st European Signal Processing Conference (EUSIPCO)*, 09 2023, pp. 191–195.
- [9] K. Wilkinghoff, T. Fujimura, K. Imoto, J. L. Roux, Z.-H. Tan, and T. Toda, “Handling domain shifts for anomalous sound detection: A review of dcase-related work,” *arXiv preprint arXiv:2503.10435*, 2025.
- [10] NTT Corporation, “DCASE 2025 challenge task 2 and DCASE 2023 challenge task 2 baseline auto encoder: dcase2023\_task2\_baseline\_ae,” GitHub Repository, 2024. [Online]. Available: [https://github.com/nttcs/nttcs-dcase2023\\_task2\\_baseline\\_ae](https://github.com/nttcs/nttcs-dcase2023_task2_baseline_ae)
- [11] S.-M. Kim and Y. Kim, “Enhancing sound-based anomaly detection using deep denoising autoencoder,” *IEEE Access*, vol. PP, pp. 1–1, 01 2024.
- [12] D. Hendrycks, M. Mazeika, and T. Dietterich, “Deep anomaly detection with outlier exposure,” *Proceedings of the International Conference on Learning Representations*, 2019.
- [13] Y. Tachioka, “Outlier exposure with efficient division of positive and negative examples for anomalous sound detection,” in *2024 32nd European Signal Processing Conference (EUSIPCO)*, 2024, pp. 76–80.
- [14] K. Wilkinghoff, “Self-supervised learning for anomalous sound detection,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 276–280.
- [15] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, “Domain generalization: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4396–4415, 2023.
- [16] J. Zhao, “Anomalous sound detection based on convolutional neural network and mixed features,” *Journal of Physics: Conference Series*, vol. 1621, p. 012025, 08 2020.
- [17] T. Nishida, K. Dohi, T. Endo, M. Yamamoto, and Y. Kawaguchi, “Anomalous sound detection based on machine activity detection,” in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 269–273.
- [18] R. Tanaka, K. Ikeda, S. Aoyama, and S. Tamura, “Anomalous sound detection using cnn-based models and ensemble,” *DCASE2023 Challenge*, Tech. Rep., June 2023.
- [19] F. Zhang, X. Xie, and K. Guo, “Asd-diffusion: Anomalous sound detection with diffusion models,” in *International Conference on Pattern Recognition*. Springer, 2025, pp. 343–355.
- [20] T. Peng, R. Qiu, J. Zhu, Y. Xiao, S. Wang, Y. Zhang, C. Zhu, S. Li, and X. Shao, “Unsupervised abnormal sound detection based on spectral coherence and feature fusion in domain displacement condition,” *DCASE2022 Challenge*, Tech. Rep., July 2022.
- [21] B. C. Chan and C. L. Lu, “An ensemble approach for abnormal sound detection with data augmentation,” *DCASE2021 Challenge*, Tech. Rep., July 2021.
- [22] X. Zheng, A. Jiang, B. Han, Y. Qian, P. Fan, J. Liu, and W.-Q. Zhang, “Improving anomalous sound detection via low-rank adaptation fine-tuning of pre-trained audio models,” in *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2024, pp. 969–974.
- [23] J. Bai, J. Chen, M. Wang, M. S. Ayub, and Q. Yan, “Ssdpt: Self-supervised dual-path transformer for anomalous sound detection,” *Digital Signal Processing*, vol. 135, p. 103939, 2023.
- [24] A. Gu and T. Dao, “Mamba: Linear-time sequence modeling with selective state spaces,” in *First Conference on Language Modeling*, 2024. [Online]. Available: <https://openreview.net/forum?id=tEYskw1VY2>
- [25] B. Sun, Y. Zhang, S. Jiang, and Y. Fu, “Hybrid pixel-unshuffled network for lightweight image super-resolution,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 2, 2023, pp. 2375–2383.
- [26] J. Guan, J. Tian, Q. Zhu, F. Xiao, H. Zhang, and X. Liu, “Disentangling hierarchical features for anomalous sound detection under domain shift,” *arXiv preprint arXiv:2501.01604*, 2025.
- [27] M. H. Erol, A. Senocak, J. Feng, and J. S. Chung, “Audio mamba: Bidirectional state space model for audio representation learning,” *IEEE Signal Processing Letters*, 2024.
- [28] R. Jiang, Z. Yang, and J. Zhao, “A complete deep support vector data description for one class learning,” *IEEE Access*, vol. 11, pp. 117 494–117 507, 2023.
- [29] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, “MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task,” in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022.
- [30] S. Falkner, A. Klein, and F. Hutter, “Bohb: Robust and efficient hyperparameter optimization at scale,” in *International conference on machine learning*. PMLR, 2018, pp. 1437–1446.
- [31] Y. Gong, Y.-A. Chung, and J. Glass, “Ast: Audio spectrogram transformer,” in *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (Interspeech 2021)*, 08 2021, pp. 571–575.