CNC: Cross-modal Normality Constraint for Unsupervised Multi-class Anomaly Detection

Xiaolei Wang[1,2,3*], **Xiaoyang Wang**[1,2,3*], **Huihui Bai**[4], **Eng Gee Lim**[1], **Jimin Xiao**[1†]
[1]Xi'an Jiaotong-Liverpool University
[2]University of Liverpool
[3]Dinnar Automation Technology
[4]Beijing Jiaotong University
{Xiaolei.Wang, wangxy}@liverpool.ac.uk, hhbai@bjtu.edu.cn, {enggee.lim, jimin.xiao}@xjtlu.edu.cn

Abstract

Existing unsupervised distillation-based methods rely on the differences between encoded and decoded features to locate abnormal regions in test images. However, the decoder trained only on normal samples still reconstructs abnormal patch features well, degrading performance. This issue is particularly pronounced in unsupervised multi-class anomaly detection tasks. We attribute this behavior to 'over-generalization' (OG) of decoder: the significantly increasing diversity of patch patterns in multi-class training enhances the model generalization on normal patches, but also inadvertently broadens its generalization to abnormal patches. To mitigate 'OG', we propose a novel approach that leverages class-agnostic learnable prompts to capture common textual normality across various visual patterns, and then apply them to guide the decoded features towards a 'normal' textual representation, suppressing 'over-generalization' of the decoder on abnormal patterns. To further improve performance, we also introduce a gated mixture-of-experts module to specialize in handling diverse patch patterns and reduce mutual interference between them in multi-class training. Our method achieves competitive performance on the MVTec AD and VisA datasets, demonstrating its effectiveness.

Code — https://github.com/cvddl/CNC

Introduction

Visual anomaly detection (AD) mainly focuses on identifying unexpected patterns (deviating from our familiar normal ones) within samples. Industrial defect detection is one of the most widely used branches of AD (Bergmann et al. 2019), which requires models to automatically recognize various defects on the surface of industrial products, such as scratches, damages, and misplacement. Due to the inability to fully collect and annotate anomalies, unsupervised methods (Yu et al. 2021; Gudovskiy, Ishizaka, and Kozuka 2022; Liu et al. 2023b) become mainstream solutions for AD. Previous unsupervised methods mostly train one model for one class of data, which requires large parameter storage and long training time as the number of classes increases. Therefore, UniAD (You et al. 2022) proposed a challenging multi-class AD setting, i.e., training one model to detect anomalies from multiple categories.

Reverse distillation (RD) (Deng and Li 2022) is a highly effective unsupervised AD (UAD) method. It employs a learnable decoder (student network) to reconstruct features from a pre-trained encoder (teacher network) on normal samples via patch-level cosine distance minimization. Ideally, the learned decoder should only recover the encoded normal patches, while failing to reconstruct unseen abnormal patterns. Anomaly regions are then detected by comparing features before and after decoding. In multi-class training, a single model is optimized on the normal samples from multiple classes to achieve unified detection. While the increasing training diversity can improve model generalization on reconstructing normal patches, it also leads to undesired generalization to unseen abnormal patch patterns. Consequently, abnormal regions are recovered well during inference, narrowing the difference between encoded and decoded features and degrading detection performance (Fig. 1(B) I.). We term this issue 'over-generalization' (OG). The key question remains: *How can we effectively mitigate 'OG' while preserving the generalization on normal samples in the multi-class distillation framework?*

To address this challenge, we seek to incorporate an additional constraint in the decoding process. Leveraging insights from vision-language models (VLMs) (Radford et al. 2021), we observe that normal and abnormal regions within a sample exhibit distinct responses to the same text description, as illustrated in Fig. 1(A). We propose to exploit this distinction in cross-modal response to differentiate the decoding of normal and abnormal patch features, thereby hindering the recovery of abnormal patterns. Specifically, we employ class-agnostic learnable prompts to extract the common normality from encoded visual features across different classes. These prompts serve as anchors in the textual space, aligning the decoded normal features with a universal representation of normality and suppressing the 'over-generalization' of the decoder towards abnormal patterns (Fig. 1(B) II.). We also design a normality promotion mechanism for feature distillation, introducing cross-modal activation as a control coefficient on visual features to increase sensitivity to unexpected abnormal patterns. We term the

*These authors contributed equally.
†Corresponding author.

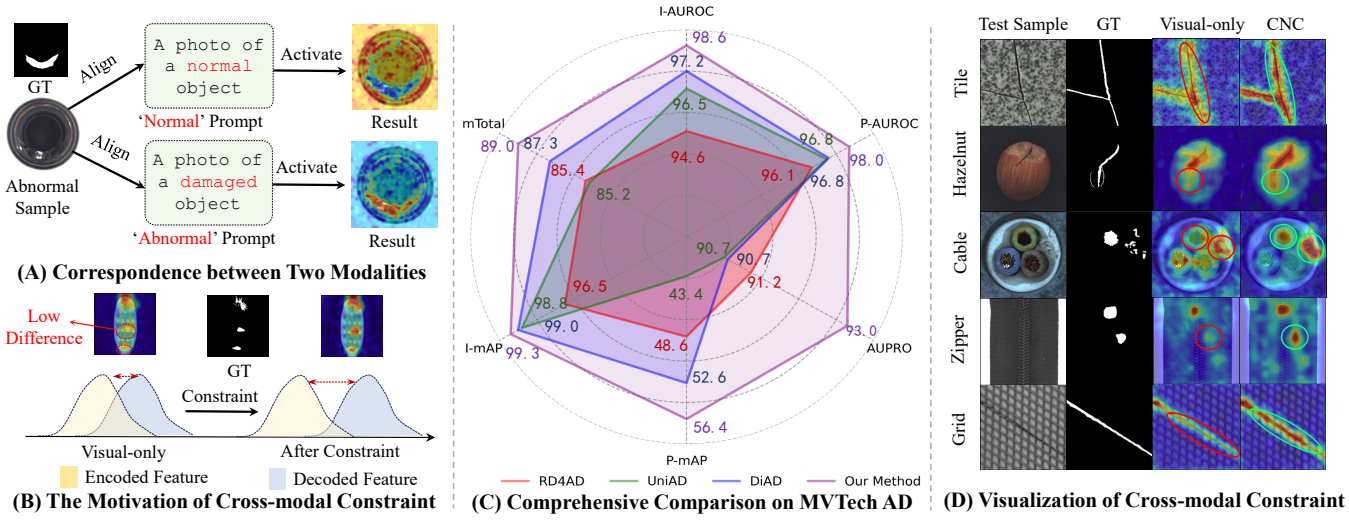


Figure 1: (A) and (B) show the correspondence between visual and text modality and the motivation of CNC, respectively. (C) shows a comprehensive performance comparison with previous SOTA methods that only learn sample normality on the visual modality on MVTec AD dataset. (D) gives some visualization results of cross-modal constraint on MVTec AD dataset.

combination of these two strategies Cross-modal Normality Constraint (CNC), which aims to mitigate the 'OG' issue and enhance anomaly localization in multi-class distillation frameworks (see Fig. 1(D) for visualization results).

Another angle to tackle the 'OG' issue is to mitigate the mutual interference among different patch patterns produced by increasing categories during feature distillation learning. The success of previous one-model-one-class settings can be attributed to the separate learning of patterns for each class, without interference from patch patterns from other classes. However, in multi-class training, inter-class interference is unavoidable. To address this issue, we propose constructing multiple expert networks to specialize in handling different patch patterns. We find that the mixture-of-experts (MoE) framework (Shazeer et al. 2017; Ma et al. 2018) can selectively process distinct patch patterns, assigning each patch a distinct weighted combination of experts to alleviate the mutual interference. By combining a vanilla RD framework with our CNC and MoE, we achieve performances that surpass previous methods (Deng and Li 2022; You et al. 2022; He et al. 2024b) across multiple metrics (see Fig. 1(C)).

Our work primarily addresses the inherent 'over-generalization' issue for distillation frameworks in multi-class training. To this end, we propose two key strategies: cross-modal normality constraint to facilitate visual decoding, and a mixture-of-experts (MoE) module to process diverse patch patterns selectively. We conduct comprehensive experiments to demonstrate the efficacy of our approach, yielding a notable performance gain over single-modal methods. The main contributions of this work are summarized as follows:

- We identify the 'OG' issue in multi-class distillation frameworks and propose a two-pronged solution to address this challenge.
- We introduce a cross-modal normality constraint to guide visual decoding, effectively reducing the effect of 'OG'.
- We design a gated MoE module to selectively handle various patch patterns, mitigating inter-pattern interference and enhancing detection performance.
- Our novel cross-modal distillation framework, built from scratch, achieves competitive performance on the MVTec AD and VisA datasets.

Related Work

Anomaly Detection

Visual anomaly detection contains various settings according to specific engineering requirements, e.g., unsupervised AD (Yi and Yoon 2020; Zou et al. 2022; Gu et al. 2023; Cao, Zhu, and Pang 2023; Liu et al. 2023a; Zhang, Xu, and Zhou 2024), zero and few-shot AD (Huang et al. 2022; Fang et al. 2023; Lee and Choi 2024), noisy AD (Chen et al. 2022; Jiang et al. 2022), and 3D AD (Gu et al. 2024; Costanzino et al. 2024; Liu et al. 2024; Li et al. 2024a). Existing unsupervised AD methods can be roughly divided into reconstruction-based (Tien et al. 2023; Lu et al. 2023b; He et al. 2024b), feature-embedding-based (McIntosh and Albu 2023; Roth et al. 2022; Lei et al. 2023), augmentation-based (Zavrtanik, Kristan, and Skočaj 2021; Zhang, Xu, and Zhou 2024; Lin and Yan 2024) methods.

Reconstruction-based Method The reconstruction-based method employs the autoencoder framework to learn the normality of training samples by reconstructing data or its features. Therefore, some works (Zhang et al. 2023; Guo et al. 2024) rethink RD as reconstruction-based method. Although this type of approach offers fast inference speed, the anomaly localization is inevitably degraded by 'OG', which is attributed to the increasing diversity of patch patterns in multi-class training. To address the issue, we propose CNC and MoE to alleviate undesired generalization.

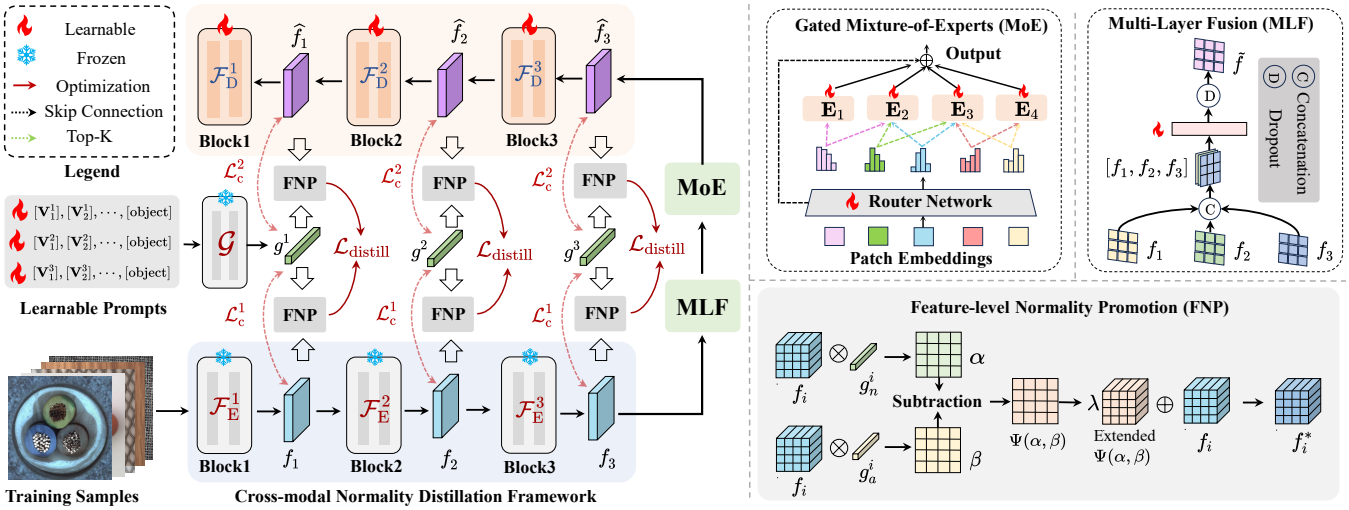


Figure 2: Overview of the proposed cross-modal normality distillation framework. Additionally, details of feature-level normality promotion, multi-layer fusion, and gated mixture-of-experts are illustrated in this graph.

Feature-embedding-based Method These methods typically rely on pre-trained networks to extract feature embedding vectors from a feature extractor trained on natural images, then apply density estimation (Defard et al. 2021; Yao et al. 2023), memory bank (Bae, Lee, and Kim 2023), etc., to detect anomalies. However, there is a significant gap between industrial and natural data, and the extracted embedding may not be suitable for anomaly detection tasks.

Augmentation-based Method Early augmentation methods (Li et al. 2021; Lu et al. 2023a) typically rely on simple handcraft augmentation techniques such as rotation, translation, and texture pasting to generate pseudo samples for discrimination training. However, a huge gap exists between the obtained synthesized samples and the real defect samples. Therefore, some works based on the generative model are proposed recently, such as VAE-based (Lee and Choi 2024), GAN-based (Wang et al. 2023), diffusion-based (Hu et al. 2024; Zhang, Xu, and Zhou 2024) methods. Due to the incomplete collection of defect shapes and categories, it is impossible to synthesize all types of anomalies.

Preliminaries

CLIP Contrastive Language Image Pre-training (Radford et al. 2021) (CLIP) is a large-scale vision-language model famous for its multi-modal alignment ability via training with a lot of image-prompt pair data. Specifically, given an unknown image $\mathbf{x}$ and text-prompts $\{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_J\}$, CLIP can predict the probability of alignment between $\mathbf{x}$ and every prompt $\mathbf{p}_j$ as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp(\mathcal{F}(\mathbf{x}) \cdot \mathcal{G}(\mathbf{p}_\mathbf{y})/\tau)}{\sum_{j=1}^{J} \exp(\mathcal{F}(\mathbf{x}) \cdot \mathcal{G}(\mathbf{p}_j)/\tau)}, \quad (1)$$

where $\mathcal{F}(\cdot)$ and $\mathcal{G}(\cdot)$ are CLIP visual and text encoder, respectively, and τ is a temperature hyperparameter. Previous work (Jeong et al. 2023) adopts handcraft prompts, such as `a photo of normal/damaged [class]`, to achieve zero-shot anomaly detection.

Prompt Learning Prompt learning (Jia et al. 2022; Zhou et al. 2022) focuses on optimizing the input prompts to enhance the language or multi-modal model performance on specific tasks. CoOp (Zhou et al. 2022) introduces a learnable prompt, $\mathbf{p} = [\mathbf{V}_1], [\mathbf{V}_2], \cdots, [\mathbf{V}_\mathrm{M}], [\mathtt{class}]$, to achieve few-shot classification, where each $[\mathbf{V}_m]$ is a learnable token, and M is the number of tokens. However, in multi-class UAD task, we do not expect to utilize any class information. Following works (Zhou et al. 2023; Li et al. 2024b), our applied learnable prompts are defined as:

$$\mathbf{p}_n = [\mathbf{V}_1^n], [\mathbf{V}_2^n], \cdots, [\mathbf{V}_\mathrm{M}^n], [\mathtt{object}], \quad (2)$$

$$\mathbf{p}_a = [\mathbf{V}_1^a], [\mathbf{V}_2^a], \cdots, [\mathbf{V}_\mathrm{M}^a], [\mathtt{damaged}], [\mathtt{object}], \quad (3)$$

where $\mathbf{p}_n$ and $\mathbf{p}_a$ are normal and abnormal prompts respectively. We argue that utilizing category-agnostic prompts enables the learning of common normality patterns across samples from different classes. Different from previous methods, we apply these prompts to learn the normality of training samples and leverage them to guide visual decoding.

Methodology

Overview

The framework of the proposed method is shown in Fig. 2. Our method consists of three main sections: (1) Visual Distillation Framework; (2) Cross-modal Normality Constraint; (3) Gated Mixture-of-Experts. In (1), we introduce our basic reverse distillation network. In (2), we propose cross-modal normality constraint to ensure decoded features to align with a textual 'normal' representation. Additionally, we propose a cross-modal control coefficient on the visual feature to improve sensitivity to abnormal patch patterns, which is called feature-level normality promotion. In (3), a gated mixture-of-experts (MoE) module is shown in detail to specifically

handle various patch patterns. Finally, the inference phase of our method is given for convenience.

Visual Distillation Framework

Compared with previous single-modal distillation-based methods (Deng and Li 2022; Tien et al. 2023), we select multi-modal backbone, CLIP-ViT, as encoder, which means that text-modal information can be adopted to improve detection performance. Specifically, for a given image $\mathbf{x} \in \mathbb{R}^{H_0 \times W_0 \times 3}$, the CLIP visual encoder $\mathcal{F}$ encodes $\mathbf{x}$ into multi-layer latent space features as $\{f_i\}_{i=1}^{N}$, where $f_i \in \mathbb{R}^{H \times W \times C}$ represents i-th layer feature (feature size in each layer of ViT is consistent), and N is the number of residual attention layers in ViT (Dosovitskiy et al. 2020). We select i_1-th, i_2-th, i_3-th layer features as visual encoded features. For convenience, let $\mathcal{F}_{\mathrm{E}}^1$, $\mathcal{F}_{\mathrm{E}}^2$, $\mathcal{F}_{\mathrm{E}}^3$ denote i_1-th, i_2-th, i_3-th blocks respectively, and f_1, f_2, f_3 is the corresponding encoded features. For visual decoder, we adopt three general residual attention blocks (the basic module of ViT) together as decoder and extract corresponding features to reconstruct. Therefore, we denote decoder as $\mathcal{F}_{\mathrm{D}}$ with three blocks $\{\mathcal{F}_{\mathrm{D}}^i\}_{i=1}^3$, with corresponding decoded features as $\{\widehat{f}_i\}_{i=1}^3$ (see Fig. 2). In addition, to further alleviate the 'over-generalization', Gaussian noise is applied on the encoded feature f_i to obtain its perturbed version f_i^{noise}.

Cross-modal Normality Constraint

To alleviate the unexpected 'over-generalization' in multi-class training, we propose a text-modal normality constraint to guide the decoded features towards a 'normal' textual representation, suppressing the 'over-generalization' of the decoder towards the abnormal direction. The key to our proposed cross-modal normality constraint lies in applying learnable category-agnostic prompts to learn common normality from various normal samples and maintain the semantic consistency of visual encoded and decoded features in textual space during the training phase.

Learning Cross-modal Normality In this section, we aim to apply class-agnostic learnable prompts to learn textural normality from various encoded features. Specifically, for a given image $\mathbf{x}$, we apply $\{\mathcal{F}_{\mathrm{E}}^i\}_{i=1}^3$ blocks in encoder to obtain multiple layer features $\{f_i\}_{i=1}^3$. According to Eq. (2) and Eq. (3), we initialize three sets of learnable prompts $\{[\mathbf{p}_n^i, \mathbf{p}_a^i]\}_{i=1}^3$, where $\{\mathbf{p}_n^i\}_{i=1}^3$ are applied to learn textual normality from different layer visual encoded features. Then, each prompt pair $[\mathbf{p}_n^i, \mathbf{p}_a^i]$ is input to CLIP text encoder $\mathcal{G}(\cdot)$, producing the corresponding text feature $[g_n^i, g_a^i]$, where $g_n^i = \mathcal{G}(\mathbf{p}_n^i)$, $g_a^i = \mathcal{G}(\mathbf{p}_a^i)$. Next, we employ a modal-alignment optimization object to learn the textual normality from $\{f_i\}_{i=1}^3$:

$$\mathcal{L}_{\mathrm{c}}^1 = \sum_{i=1}^{3} -\log \frac{\exp(\mathbf{e}_i \cdot g_n^i/\tau)}{\exp(\mathbf{e}_i \cdot g_n^i/\tau) + \exp(\mathbf{e}_i \cdot g_a^i/\tau)}, \quad (4)$$

where $\mathbf{e}_i$ represents the global feature of f_i, τ is a temperature coefficient. Next, textual features $\{[g_n^i, g_a^i]\}_{i=1}^3$ are adopted in feature distillation and decoding to alleviate unexpected 'over-generalization'.

Feature Distillation with Normality Promotion In this section, textual features $\{[g_n^i, g_a^i]\}_{i=1}^3$ are applied on distillation to improve sensitivity to anomaly patch patterns. Specifically, we first define a cross-modal control coefficient on visual encoded and decoded features f_i and $\widehat{f}_i$. It is designed to compute a cross-modal activation map between visual patch features and text features g_n^i, improving sensitivity to unexpected abnormal patch patterns. Specifically, we design the new encoded feature f_i^* with a cross-modal normality control coefficient as follows:

$$f_i^* = f_i \oplus \lambda \Psi(\alpha_i, \beta_i), \quad (5)$$

where $f_i, f_i^* \in \mathbb{R}^{H \times W \times C}$, $\lambda = 1/\|f_i\|$ is a scaled coefficient, $\|\cdot\|$ is the $\mathbf{L}_2$ norm, and control coefficient $\Psi(\alpha, \beta)$ is written as

$$\Psi(\alpha_i, \beta_i) = \frac{1}{2}(1 + \mathbf{tanh}(\alpha_i - \beta_i)), \quad (6)$$

where weight maps α and β are defined as:

$$\alpha_i = f_i \otimes g_n^i, \beta_i = f_i \otimes g_a^i, \quad (7)$$

where $g_n^i \in \mathbb{R}^{1 \times 1 \times C}$, $\alpha_i, \beta_i \in \mathbb{R}^{H \times W}$, $\otimes$ denotes the vector-wise product between g_n^i and each patch embedding $\mathbf{z}_i^l$ of f_i, l is the index of patch embedding. Therefore, $\Psi(\alpha_i, \beta_i) \in \mathbb{R}^{H \times W}$, and $\oplus$ is element-wise addition operation. Similarly, following Eq. (5) and Eq. (6), we also obtain the decoded feature $\widehat{f}_i^*$ with cross-modal control coefficient:

$$\widehat{f}_i^* = \widehat{f}_i \oplus \lambda \Psi(\widehat{\alpha}_i, \widehat{\beta}_i), \quad (8)$$

where $\widehat{\alpha}_i = \widehat{f}_i \otimes g_n^i$, $\widehat{\beta}_i = \widehat{f}_i \otimes g_a^i$.

We call the above step 'feature-level normality promotion' (FNP), as shown in Fig. 2. Next, we give a new cross-modal distillation loss to ensure the consistency of the encoded and decoded features with the corresponding control coefficient as follows:

$$\mathcal{L}_{\mathrm{distill}} = \sum_{i=1}^{3} \Big(1 - \frac{\mathbf{Flat}(f_i^*) \cdot \mathbf{Flat}(\widehat{f}_i^*)}{\|\mathbf{Flat}(f_i^*)\| \|\mathbf{Flat}(\widehat{f}_i^*)\|}\Big), \quad (9)$$

where $\mathbf{Flat}(\cdot)$ is the fatten function.

Feature Decoding with Normality Constraint In this section, we apply textual features $\{[g_n^i, g_a^i]\}_{i=1}^3$ trained by (4) as anchors to guide feature decoding, alleviating unexpected 'OG'. Our solution is to constrain the textual representation of the decoded features not to deviate from 'normal' during training, i.e., we also keep class tokens of decoded features aligning with normal text features $\{g_n^i\}$:

$$\mathcal{L}_{\mathrm{c}}^2 = \sum_{i=1}^{3} -\log \frac{\exp(\widehat{\mathbf{e}}_i \cdot g_n^i/\tau)}{\exp(\widehat{\mathbf{e}}_i \cdot g_n^i/\tau) + \exp(\widehat{\mathbf{e}}_i \cdot g_a^i/\tau)}, \quad (10)$$

where $\widehat{\mathbf{e}}_i$ represents the global feature of $\widehat{f}_i$. We combine formulas (4) and (10) to give the cross-modal constraint loss:

$$\mathcal{L}_{\mathrm{constraint}} = \begin{cases} \mathcal{L}_{\mathrm{c}}^1 & \text{if epoch} < \vartheta \\ \mathcal{L}_{\mathrm{c}}^1 + \gamma \mathcal{L}_{\mathrm{c}}^2 & \text{if epoch} \geq \vartheta \end{cases}, \quad (11)$$

where $\gamma = 0.1$ is a hyperparameter. Each text feature g_n^i is a dynamic anchor that is used as a medium to keep the decoded feature toward the 'normal' direction.

Gated Mixture-of-Experts Module

Multi-Layer Fusion Following works (Deng and Li 2022; Tien et al. 2023), different layer features of pre-trained encoder are aggregated, improving detection performance. For an input $\mathbf{x}$, we first apply encoder $\mathcal{F}_{\mathrm{E}}$ to extract multi-layer features $\{f_i\}_{i=1}^3$, and concatenate them as $[f_1, f_2, f_3]$. We adopt a projection layer $\Phi(\cdot)$ (a linear layer with dropout block) to transfer its channel dimension to C:

$$\widetilde{f} = \Phi([f_1, f_2, f_3]), \tag{12}$$

where $\widetilde{f} \in \mathbb{R}^{H\times W\times C}$ is a fusion feature (see Fig. 2 MLF) and is input to the following MoE module.

Gated Mixture-of-Experts We employ different expert combination to handle different patch patterns, reducing the mutual interference between them. Specifically, for a given mini-batch of fusion features, we obtain a batch of patch embedding features $\{\mathbf{z}_r\}_{r=1}^R$, where $R = B * H * W$. Our goal is to assign different expert combinations to recognize different patch patterns. Therefore, we first use a router network $\mathbf{G}(\cdot)$ to assign an expert-correlation score to each patch, i.e.,

$$\mathbf{H}_t = \mathbf{G}_t(\mathbf{z}_r), \;\; t \in \{1, 2, \cdots, T\}, \tag{13}$$

where $\mathbf{G}(\cdot) : \mathbb{R}^C \longmapsto \mathbb{R}^T$, T is the number of expert networks $\{\mathbf{E}_t(\cdot)\}_{t=1}^T$, and each $\mathbf{E}_i(\cdot)$ is conducted by a MLP. Next, we select experts with top K scores $\{\mathbf{H}_k\}_{k=1}^K$ to handle the patch embedding vector $\mathbf{z}_r$, denoted by $\{\mathbf{E}_k(\cdot)\}_{k=1}^K$. Next, we obtain a unique combination of processes on each patch embedding, i.e.,

$$\mathbf{z}_r^* = \sum_{k=1}^{K} \mathbf{H}_k * \mathbf{E}_k(\mathbf{z}_r). \tag{14}$$

To prevent the router from assigning dominantly large weights to a few experts, which can lead to a singular scoring operation, we apply a universal importance loss (Bengio et al. 2015) to optimize the MoE:

$$\mathcal{L}_{\mathrm{moe}} = \frac{\mathbf{SD}(\sum_{r=1}^R \mathbf{G}(\mathbf{z}_r))^2}{(\frac{1}{R}\sum_{r=1}^R \mathbf{G}(\mathbf{z}_r))^2 + \varepsilon}, \tag{15}$$

where $\mathbf{SD}(\cdot)$ is standard deviation operation, ε is added for numerical stability. Finally, according to Eq. (9), (11), and (15), we obtain a total loss to train our model:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{distill}} + \mathcal{L}_{\mathrm{constraint}} + \mathcal{L}_{\mathrm{moe}}. \tag{16}$$

Inference

The inference is consistent with the training phase. We apply encoder $\mathcal{F}_{\mathrm{E}}(\cdot)$, learned prompts $\{[\mathbf{p}_n^i, \mathbf{p}_a^i]\}_{i=1}^3$, multi-layer fusion $\Phi(\cdot)$, MoE module, and decoder $\mathcal{F}_{\mathrm{D}}(\cdot)$ to produce encoded features $\{f_i^*\}_{i=1}^3$ and decoded features $\{\widehat{f}_i^*\}_{i=1}^3$. We design a pixel-level anomaly score map as:

$$\mathcal{S}(f_i^*, \widehat{f}_i^*) = \sum_{i=1}^{3} \sigma_i\Big(1 - \mathbf{d}(f_i^*, \widehat{f}_i^*)\Big), \tag{17}$$

where $\mathbf{d}(\cdot,\cdot)$ is pixel-wise cosine similarity, $\sigma(\cdot)$ is the upsampling factor in order to keep the same size as the input image. In addition, the image-level anomaly score is defined as the maximum score of the pixel-level anomaly map.

Experiments

Experimental Setup

Datasets MVTec AD (Bergmann et al. 2019) is the most widely used industrial anomaly detection dataset, containing 15 categories of sub-datasets. The training set consists of 3629 images with anomaly-free samples. The test dataset includes 1725 normal and abnormal images. Segmentation masks are provided for anomaly localization evaluation. VisA (Zou et al. 2022) is a challenging AD dataset containing 10821 images and 12 categories of sub-datasets.

Evaluation Metrics Following the prior work (He et al. 2024b), image-level Area Under the Receiver Operating Characteristic Curve (I-AUROC) and Average Precision (I-mAP) are applied for anomaly classification. Pixel-level AUROC, pixel-level mAP, and AUPRO (Bergmann et al. 2020) are used for anomaly localization.

Implementation Details The implementation is based on Pytorch. The publicly available CLIP model (VIT-L/14@336px) is the backbone of our method. We select the Adam optimizer to train our model. Then, we resize the resolution of each image to 224×224. In addition, the length of each learnable text prompt is set to 12, consistent with previous work (Zhou et al. 2023). For both datasets, we set temperature coefficient $\tau = 0.001$ and batch size to 8 with learning rate 0.001 to train the whole model. The number of experts is set to 5 with top $K = 2$ gated scores in the MoE. Next, we set the epoch to 250 and 200 for MVTec AD and VisA with the same $\vartheta = 5$, respectively. All experiments are conducted on a single NVIDIA Tesla V100 32GB GPU.

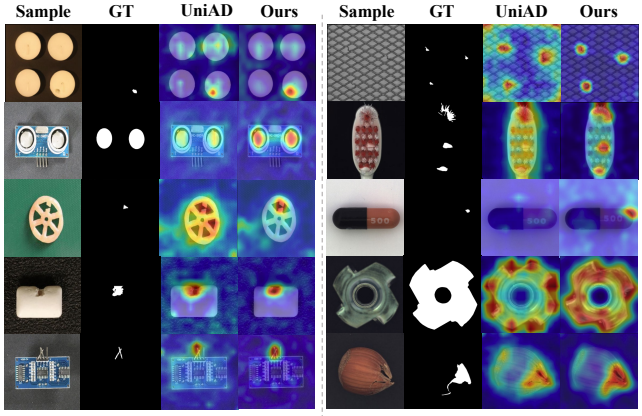
Comprehensive Comparisons with SOTA Methods

In this section, we compare our approach with several SOTA methods on MVTec AD and VisA datasets, where 5 different metrics, I-AUROC, P-AURPOC, AUPRO, I-mAP, P-mAP, are shown for comprehensive evaluation in Table 1 and Table 3, respectively.

Results on MVTec AD As reported in Table 1, for the widely used MVTec AD dataset, our cross-modal normality distillation framework (CND) achieves five different metrics by $98.6/98.0/93.0/99.3/56.4$, and the mean performance of five metrics by 89.0 under multi-class setting. Compared to UniAD and DiAD, our method improves five metrics by $+2.1/1.2/2.3/0.5/13.0$ and $+1.4/1.2/2.3/0.3/3.8$, and the mean metric by $+3.8$ and $+1.7$, respectively. Our method significantly outperforms the single-modal distillation framework, RD4AD, by $+4.0/1.9/1.9/2.8/7.8$, in terms of five metrics. In addition, our method is more stable than previous methods, achieving a performance of $93.8+$ in terms of image and pixel-level AUROC for all categories. However, RD4AD merely achieves a 60.8 I-AUROC metric on Hazelnut, and UniAD achieves a 63.1 P-AUROC metric on Grid. To further illustrate the effectiveness of our proposed method in anomaly localization, we visualize UniAD and our method prediction detection results in Fig. 3 (B).

Method ⟶ / Category ↓	**RD4AD*** (Deng and Li 2022) **CVPR2022**	**UniAD*** (You et al. 2022) **NeurIPS2022**	**DiAD** (He et al. 2024b) **AAAI2024**	**CND** **Ours**
Bottle	99.6/96.1/91.1/99.9/48.6	97.0/98.1/93.1/**100.**/66.0	99.7/98.4/86.6/99.5/52.2	**100./99.0/97.1/100./81.8**
Cable	84.1/85.1/75.1/89.5/26.3	95.2/97.3/86.1/95.9/39.9	94.8/96.8/80.5/98.8/50.1	**98.9/98.2/92.5/99.3/64.1**
Capsule	94.1/**98.8/94.8**/96.9/**43.4**	86.9/98.5/92.1/97.8/42.7	89.0/97.1/87.2/97.5/42.0	**98.0**/98.2/93.8/**99.3**/36.9
Hazelnut	60.8/97.9/92.7/69.8/36.2	99.8/98.1/94.1/**100.**/55.2	99.5/98.3/91.5/99.7/**79.2**	**100./98.8/94.9/100.**/53.3
Metal Nut	**100.**/94.8/**91.9/100.**/55.5	99.2/62.7/81.8/99.9/14.6	99.1/**97.3**/90.6/96.0/30.0	**100.**/95.5/89.2/**100./68.4**
Pill	**97.5**/97.5/95.8/**99.6**/63.4	93.7/95.0/95.3/98.7/44.0	95.7/95.7/89.0/98.5/46.0	96.8/**98.8/96.1**/99.5/**80.2**
Screw	**97.7/99.4/96.8**/99.3/40.2	87.5/98.3/95.2/96.5/28.7	90.7/97.9/95.0/**99.7/60.6**	93.8/99.0/94.7/98.3/26.2
Toothbrush	97.2/**99.0**/92.0/99.0/53.6	94.2/98.4/87.9/97.4/34.9	**99.7/99.0/95.0/99.9/78.7**	99.5/**99.0**/93.0/99.8/49.7
Transistor	94.2/85.9/74.7/95.2/42.3	**99.8/97.9/93.5**/98.0/**59.5**	**99.8**/95.1/90.0/**99.0**/15.6	96.0/94.5/74.1/94.4/57.3
Zipper	**99.5/98.5/94.1/99.9**/53.9	95.8/96.8/92.6/99.5/40.1	99.3/96.2/91.6/99.8/**60.7**	99.1/97.6/93.8/99.8/45.1
Carpet	98.5/99.0/95.1/99.6/58.5	99.8/98.5/94.4/**99.9**/49.9	99.4/98.6/90.6/**99.9**/42.2	**99.9/99.3/97.0/99.9/70.7**
Grid	98.0/96.5/**97.0**/99.4/23.0	98.2/63.1/92.9/99.5/10.7	98.5/96.6/94.0/**99.8/66.0**	**99.1/98.4**/95.0/99.7/25.8
Leather	**100.**/99.3/97.4/**100.**/38.0	**100.**/98.8/96.8/**100.**/32.9	99.8/98.8/91.3/99.7/**56.1**	**100./99.5/98.7/100.**/50.1
Tile	98.3/95.3/85.8/99.3/48.5	99.3/91.8/78.4/99.8/42.1	96.8/92.4/90.7/99.9/65.7	**100./97.7/94.1/100./73.4**
Wood	99.2/95.3/90.0/99.8/47.8	98.6/93.2/86.7/99.6/37.2	**99.7**/93.3/**97.5/100.**/43.3	98.3/**96.4**/91.4/99.5/**63.3**
Mean	94.6/96.1/91.1/96.5/48.6	96.5/96.8/90.7/98.8/43.4	97.2/96.8/90.7/99.0/52.6	**98.6/98.0/93.0/99.3/56.4**
mTotal	85.4	85.2	87.3	**89.0**

Table 1: Comprehensive anomaly detection results with image-level AUROC, pixel-level AUROC, AUPRO, image-level mAP, and pixel-level mAP metrics on **MVTec AD** dataset. We also provide the average of all metrics at the bottom of the table. The best and second-best results are in bold and underlined, respectively. *: The results are sourced from (He et al. 2024a).



(A) Examples on VisA **(B) Examples on MVTec AD**

Figure 3: Visualization for detection results of UniAD and our method on MVTec AD and VisA datasets.

	MLF	CNC	MoE	Performance	Mean
i	✗	✗	✗	95.4/95.6/90.3	93.7
ii	✓	✗	✗	96.0/96.3/91.1	94.4
iii	✗	✓	✗	96.7/97.0/92.0	95.2
iv	✗	✗	✓	96.3/95.8/90.7	94.2
v	✓	✓	✗	98.1/97.7/92.8	96.2
vi	✓	✗	✓	97.0/96.8/91.4	95.0
vii	✓	✓	✓	**98.6/98.0/93.0**	**96.5**

Table 2: Ablation study of our method on MVTec AD. MLF: Multi-Layer Fusion. CNC: Cross-modal Normality Constraint, including feature-level normality promotion, constraint loss $\mathcal{L}_{\text{constraint}}$, and distillation loss $\mathcal{L}_{\text{distill}}$. MoE: Gated Mixture-of-Expert Module. **Bold**/underline values indicate the best/runner-up.

Results on VisA As reported in Table 3, for the VisA dataset, our proposed method also achieves a SOTA performance, namely 93.2 and 98.5 in terms of I-AUROC and P-AUROC metrics, respectively. Compared to previous multi-class methods, our method outperforms UniAD by $+7.7/2.6/15.8/7.1/16.8$, and DiAD by $+6.4/2.5/16.2/4.3/11.7$. Especially, in the 'fryum' and 'capsules' categories, our method greatly improves anomaly classification compared to UniAD (You et al. 2022) and (He et al. 2024b). Finally, we also visualize our obtained localization result via heat maps on VisA in Fig. 3 (A).

Ablation Study

In this section, we investigate the contribution of different main components in our approach. Additionally, we show results on different backbones with different resolutions and investigate the impact of hyperparameters of MoE.

Effectiveness of Main Components In Table 2, we report three main metrics, including I-AUROC, P-AUROC and AUPRO, to study the impact of each key component on MVTec AD. As shown in line i and iii of Table 2, CNC improves our base model by $+1.3/1.4/1.7$. In addition, equipped with 'MLF' module, we get a gain of $+2.1/1.4/1.7$ via CNC (see ii and v in Table 2). Therefore, our proposed CNC successfully suppresses undesired 'OG'. In addition, the proposed MoE module alleviates 'OG' by assigning different weights to different patch patterns, improving performance by $+1.0/0.5$ in terms of I-AUROC and P-AUROC (see ii and vi). We also found that MoE enhances image-level anomaly detection, enhancing $+0.5$ on I-AUROC metric (see v and vii in Table 2). Finally, our designed multi-layer fusion module can well fuse different

Method ⟶ / Category ↓	**RD4AD*** (Deng and Li 2022) **CVPR2022**	**UniAD*** (You et al. 2022) **NeurIPS2022**	**DiAD*** (He et al. 2024b) **AAAI2024**	**CND Ours**
pcb1	**96.2**/99.4/**95.8**/**95.5**/66.2	92.8/93.3/64.1/92.7/ 3.9	88.1/98.7/80.2/88.7/49.6	94.1/**99.5**/92.6/91.7/**70.7**
pcb2	**97.8**/98.0/**90.8**/**97.8**/**22.3**	87.8/93.9/66.9/87.7/ 4.2	91.4/95.2/67.0/91.4/ 7.5	95.9/**98.4**/88.8/92.2/18.1
pcb3	**96.4**/97.9/**93.9**/**96.2**/**26.2**	78.6/97.3/70.6/78.6/13.8	86.2/96.7/68.9/87.6/ 8.0	92.0/**98.6**/93.7/93.9/ 21.7
pcb4	**99.9**/97.8/88.7/**99.9**/31.4	98.8/94.9/72.3/98.8/14.7	99.6/97.0/85.0/99.5/17.6	**99.9**/**99.0**/**90.5**/99.8/ **40.5**
macaroni1	75.9/**99.4**/**95.3**/61.5/ 2.9	79.9/97.4/84.0/79.8/ 3.7	85.7/94.1/68.5/**85.2**/**10.2**	**86.7**/98.6/90.5/84.4/ 7.8
macaroni2	**88.3**/**99.7**/**97.4**/**84.5**/**13.2**	71.6/95.2/76.6/71.6/ 0.9	62.5/93.6/73.1/57.4/ 0.9	84.4/98.1/93.6/81.3/12.7
capsules	82.2/**99.4**/**93.1**/**90.4**/**60.4**	55.6/88.7/43.7/55.6/ 3.0	58.2/97.3/77.9/69.0/10.0	**83.4**/98.4/88. 6/89.2/33.6
candle	92.3/**99.1**/**94.9**/92.9/**25.3**	**94.1**/98.5/91.6/**94.0**/17.6	92.8/97.3/89.4/92.0/12.8	93.7/98.4/91.9/90.0/16.7
cashew	92.0/91.7/86.2/**95.8**/44.2	92.8/**98.6**/**87.9**/92.8/51.7	91.5/90.9/61.8/95.7/53.1	**94.1**/98.1/87.4/92.8/62.9
chewinggum	94.9/98.7/76.9/97.5/59.9	96.3/98.8/81.3/96.2/54.9	**99.1**/94.7/59.5/**99.5**/11.9	98.7/**99.1**/**89.4**/99.2/**61.3**
fryum	95.3/97.0/**93.4**/**97.9**/47.6	83.0/95.9/76.2/83.0/34.0	89.8/**97.6**/81.3/95.0/**58.6**	**96.4**/97.0/92.1/**97.9**/47.3
pipe-fryum	97.9/99.1/95.4/98.9/56.8	94.7/98.9/91.5/94.7/50.2	96.2/**99.4**/89.9/98.1/**72.7**	**98.9**/98.4/**97.5**/**99.0**/61.4
Mean	92.4/98.1/**91.8**/92.4/**38.0**	85.5/95.9/75.6/85.5/21.0	86.8/96.0/75.2/88.3/26.1	**93.2**/**98.5**/91.4/**92.6**/37.8
mTotal	82.5	72.7	74.5	**82.7**

Table 3: Comprehensive anomaly detection results with five different metrics on **VisA** dataset. **Bold**/underline values indicate the best/runner-up. *: The results are sourced from (He et al. 2024a).

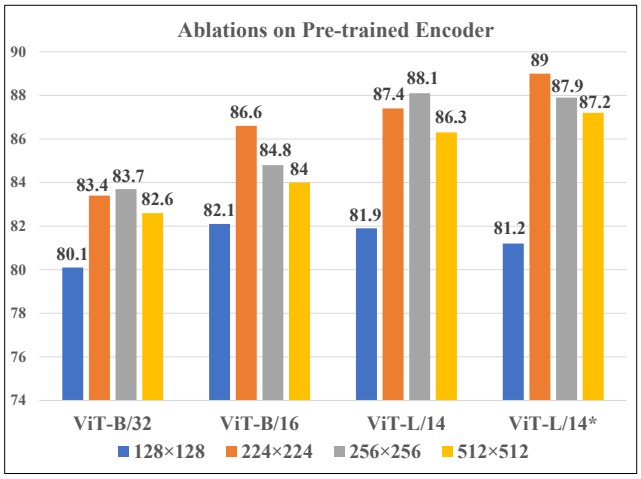


Figure 4: Choices on four pre-trained teacher network (encoder) with four different resolutions. The vertical axis represents the average value of I-AUROC/P-AUROC/AUPRO/I-mAP/P-mAP. ViT-L/14* denotes pre-trained CLIP model, VIT-L/14@336px.

layer information of ViT. As shown i and ii in Table 2, MLF improves three metrics by $+0.6/0.7/0.8$.

Choices on Pre-trained Encoders and Image Resolutions Fig. 4 shows results of different resolutions for four pre-trained encoder. On the one hand, we found that models that perform well in zero-shot classification have higher performance in our framework. We obtain the highest performance of 89.0 when applying ViT-L/14*, but the performance rapidly degraded when using ViT-B/32 and ViT-B/16. On the other hand, we found that both low and high resolutions (128×128 and 512×512) degrade detection performance and great performances can be achieved with resolutions 224×224 and 256×256, which is consistent with previous work (Deng and Li 2022).

Top K ⟶ / **No.Experts** ↓	$K=1$	$K=2$	$K=3$	$K=4$
None	98.1/97.7	✗	✗	✗
$T=1$	98.0/97.7	✗	✗	✗
$T=2$	97.4/96.7	97.2/96.9	✗	✗
$T=3$	97.4/97.0	98.0/97.3	97.5/97.5	✗
$T=4$	97.7/97.2	98.2/97.4	97.9/97.8	97.3/97.6
$T=5$	97.6/97.4	**98.6**/**98.0**	98.2/97.7	98.0/97.7
$T=6$	97.3/97.3	98.5/97.3	97.8/97.7	97.4/97.4

Table 4: Impact of Hyperparameters in MoE module, where I-AUROC and P-AUROC metrics are reported on MVTec AD dataset. T and K denote the number of experts and Top K coefficient respectively, and $K \leq T$. **Bold**/underline values indicate the best/runner-up.

Impact of Hyperparameters in MoE According to Table 4, an appropriate selection on hyperparameter greatly improves anomaly localization and classification for our method. When $T = 1$, it is equivalent to connecting an adapter and does not significantly affect performance. Both large and small values of T can degrade performance. We consider that large T may lead to some experts under-fitting and small T may result in some experts over-fitting. when $T = 5$ and $K = 2$, the best performance is achieved.

Conclusion

In this paper, we propose a cross-modal distillation framework to address the inevitable 'over-generalization' in multi-class training. Firstly, we propose cross-modal normality constraint (CNC) to guide decoded features to align the decoded features with a textual representation of normality, thereby improving the normality of the distilled features and final detection performance. We also propose a gated MoE module to re-weight different patch patterns, reducing the mutual interference between them. Finally, extensive experiments show that our method achieves competitive performance on MVTec AD and VisA datasets.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62471405, 62331003, 62301451), Jiangsu Basic Research Program Natural Science Foundation (SBK2024021981), Suzhou Basic Research Program (SYG202316) and XJTLU REF-22-01-010, XJTLU AI University Research Centre, Jiangsu Province Engineering Research Centre of Data Science and Cognitive Computation at XJTLU and SIP AI innovation platform (YZCXPT2022103).

References

Bae, J.; Lee, J.-H.; and Kim, S. 2023. PNI: Industrial anomaly detection using position and neighborhood information. In *ICCV*.

Bengio, E.; Bacon, P.-L.; Pineau, J.; and Precup, D. 2015. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*.

Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2019. MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*.

Bergmann, P.; Fauser, M.; Sattlegger, D.; and Steger, C. 2020. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *CVPR*.

Cao, T.; Zhu, J.; and Pang, G. 2023. Anomaly detection under distribution shift. In *ICCV*.

Chen, Y.; Tian, Y.; Pang, G.; and Carneiro, G. 2022. Deep one-class classification via interpolated gaussian descriptor. In *AAAI*.

Costanzino, A.; Ramirez, P. Z.; Lisanti, G.; and Di Stefano, L. 2024. Multimodal industrial anomaly detection by cross-modal feature mapping. In *CVPR*.

Defard, T.; Setkov, A.; Loesch, A.; and Audigier, R. 2021. Padim: A patch distribution modeling framework for anomaly detection and localization. In *ICPR*.

Deng, H.; and Li, X. 2022. Anomaly detection via reverse distillation from one-class embedding. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fang, Z.; Wang, X.; Li, H.; Liu, J.; Hu, Q.; and Xiao, J. 2023. Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In *ICCV*.

Gu, Z.; Liu, L.; Chen, X.; Yi, R.; Zhang, J.; Wang, Y.; Wang, C.; Shu, A.; Jiang, G.; and Ma, L. 2023. Remembering normality: Memory-guided knowledge distillation for unsupervised anomaly detection. In *CVPR*.

Gu, Z.; Zhang, J.; Liu, L.; Chen, X.; Peng, J.; Gan, Z.; Jiang, G.; Shu, A.; Wang, Y.; and Ma, L. 2024. Rethinking Reverse Distillation for Multi-Modal Anomaly Detection. In *AAAI*.

Gudovskiy, D.; Ishizaka, S.; and Kozuka, K. 2022. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *WACV*.

Guo, J.; Jia, L.; Zhang, W.; Li, H.; et al. 2024. Recontrast: Domain-specific anomaly detection via contrastive reconstruction. In *NeurIPS*.

He, H.; Bai, Y.; Zhang, J.; He, Q.; Chen, H.; Gan, Z.; Wang, C.; Li, X.; Tian, G.; and Xie, L. 2024a. Mambaad: Exploring state space models for multi-class unsupervised anomaly detection. *arXiv preprint arXiv:2404.06564*.

He, H.; Zhang, J.; Chen, H.; Chen, X.; Li, Z.; Chen, X.; Wang, Y.; Wang, C.; and Xie, L. 2024b. A diffusion-based framework for multi-class anomaly detection. In *AAAI*.

Hu, T.; Zhang, J.; Yi, R.; Du, Y.; Chen, X.; Liu, L.; Wang, Y.; and Wang, C. 2024. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *AAAI*.

Huang, C.; Guan, H.; Jiang, A.; Zhang, Y.; Spratling, M.; and Wang, Y.-F. 2022. Registration based few-shot anomaly detection. In *ECCV*.

Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; and Dabeer, O. 2023. Winclip: Zero-/few-shot anomaly classification and segmentation. In *CVPR*.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *ECCV*.

Jiang, X.; Liu, J.; Wang, J.; Nie, Q.; Wu, K.; Liu, Y.; Wang, C.; and Zheng, F. 2022. Softpatch: Unsupervised anomaly detection with noisy data. In *NeurIPS*.

Lee, M.; and Choi, J. 2024. Text-guided variational image generation for industrial anomaly detection and segmentation. In *CVPR*.

Lei, J.; Hu, X.; Wang, Y.; and Liu, D. 2023. Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow. In *CVPR*.

Li, C.-L.; Sohn, K.; Yoon, J.; and Pfister, T. 2021. Cutpaste: Self-supervised learning for anomaly detection and localization. In *CVPR*.

Li, W.; Xu, X.; Gu, Y.; Zheng, B.; Gao, S.; and Wu, Y. 2024a. Towards Scalable 3D Anomaly Detection and Localization: A Benchmark via 3D Anomaly Synthesis and A Self-Supervised Learning Network. In *CVPR*.

Li, Y.; Goodge, A.; Liu, F.; and Foo, C.-S. 2024b. PromptAD: Zero-shot anomaly detection using text prompts. In *CVPR*.

Lin, J.; and Yan, Y. 2024. A Comprehensive Augmentation Framework for Anomaly Detection. In *AAAI*.

Liu, J.; Xie, G.; Chen, R.; Li, X.; Wang, J.; Liu, Y.; Wang, C.; and Zheng, F. 2024. Real3d-ad: A dataset of point cloud anomaly detection. In *NeurIPS*.

Liu, W.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2023a. Diversity-measurable anomaly detection. In *CVPR*.

Liu, Z.; Zhou, Y.; Xu, Y.; and Wang, Z. 2023b. Simplenet: A simple network for image anomaly detection and localization. In *CVPR*.

Lu, F.; Yao, X.; Fu, C.-W.; and Jia, J. 2023a. Removing anomalies as noises for industrial defect localization. In *ICCV*.

Lu, R.; Wu, Y.; Tian, L.; Wang, D.; Chen, B.; Liu, X.; and Hu, R. 2023b. Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection. In *NeurIPS*.

Ma, J.; Zhao, Z.; Yi, X.; Chen, J.; Hong, L.; and Chi, E. H. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD*.

McIntosh, D.; and Albu, A. B. 2023. Inter-realization channels: Unsupervised anomaly detection beyond one-class classification. In *CVPR*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; and Gehler, P. 2022. Towards total recall in industrial anomaly detection. In *CVPR*.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Tien, T. D.; Nguyen, A. T.; Tran, N. H.; Huy, T. D.; Duong, S.; Nguyen, C. D. T.; and Truong, S. Q. 2023. Revisiting reverse distillation for anomaly detection. In *CVPR*.

Wang, R.; Hoppe, S.; Monari, E.; and Huber, M. F. 2023. Defect transfer gan: Diverse defect synthesis for data augmentation. *arXiv preprint arXiv:2302.08366*.

Yao, X.; Li, R.; Zhang, J.; Sun, J.; and Zhang, C. 2023. Explicit boundary guided semi-push-pull contrastive learning for supervised anomaly detection. In *CVPR*.

Yi, J.; and Yoon, S. 2020. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *ACCV*.

You, Z.; Cui, L.; Shen, Y.; Yang, K.; Lu, X.; Zheng, Y.; and Le, X. 2022. A unified model for multi-class anomaly detection. In *NeurIPS*.

Yu, J.; Zheng, Y.; Wang, X.; Li, W.; Wu, Y.; Zhao, R.; and Wu, L. 2021. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*.

Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2021. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *ICCV*.

Zhang, J.; Chen, X.; Wang, Y.; Wang, C.; Liu, Y.; Li, X.; Yang, M.-H.; and Tao, D. 2023. Exploring plain vit reconstruction for multi-class unsupervised anomaly detection. *arXiv preprint arXiv:2312.07495*.

Zhang, X.; Xu, M.; and Zhou, X. 2024. RealNet: A feature selection network with realistic synthetic anomaly for anomaly detection. In *CVPR*.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022. Learning to prompt for vision-language models. *IJCV*, 130(9): 2337–2348.

Zhou, Q.; Pang, G.; Tian, Y.; He, S.; and Chen, J. 2023. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2310.18961*.

Zou, Y.; Jeong, J.; Pemula, L.; Zhang, D.; and Dabeer, O. 2022. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *ECCV*.