arXiv:2504.05662v5 [cs.CV] 12 Mar 2026

InvAD: Inversion-based Reconstruction-Free Anomaly Detection with Diffusion Models

Shunsuke Sakai[1] Xiangteng He[2,3] Chunzhi Gu[1*] Leonid Sigal[2,3] Tatsuhito Hasegawa[1*]
[1]University of Fukui [2]University of British Columbia [3]Vector Institute for AI
sshunsuke0102@gmail.com, xiangteng.he@ubc.ca,
czgu@ieee.org, lsigal@cs.ubc.ca, t-hase@u-fukui.ac.jp

Abstract

Despite the remarkable success, recent reconstruction-based anomaly detection (AD) methods via diffusion modeling still involve fine-grained noise-strength tuning and computationally expensive multi-step denoising, leading to a fundamental tension between fidelity and efficiency. In this paper, we propose ***InvAD****, a novel* **Inv***ersion-based* **A***nomaly* **D***etection approach – "detection via noising in latent space" – which circumvents explicit reconstruction. Importantly, we contend that the limitations in prior reconstruction-based methods originate from the prevailing "detection via denoising in RGB space" paradigm. To address this, we model AD under a reconstruction-free formulation, which directly infers the final latent variable corresponding to the input image via DDIM inversion, and then measures the deviation based on the known prior distribution for anomaly scoring. Specifically, in approximating the original probability flow ODE using the Euler method, we only enforce very few inversion steps to noise the clean image to pursue inference efficiency. As the added noise is adaptively derived with the learned diffusion model, the original features for the clean testing image can still be leveraged to yield high detection accuracy. We perform extensive experiments and detailed analysis across four widely used AD benchmarks under the unsupervised unified setting to demonstrate the effectiveness of our model, achieving state-of-the-art AD performance, and about* $2\times$ *inference time speedup without diffusion distillation. Project page:* https://invad-project.com

1. Introduction

Image anomaly detection (AD) focuses on identifying images that deviate from normal patterns, with critical applications in industrial defect detection and automated medical diagnosis [1, 2]. In real-world scenarios, anomalous samples are often scarce and costly to acquire. Consequently, researchers began focusing on unsupervised AD, where the detection model was trained solely on normal images to identify potential anomalies [3, 4].

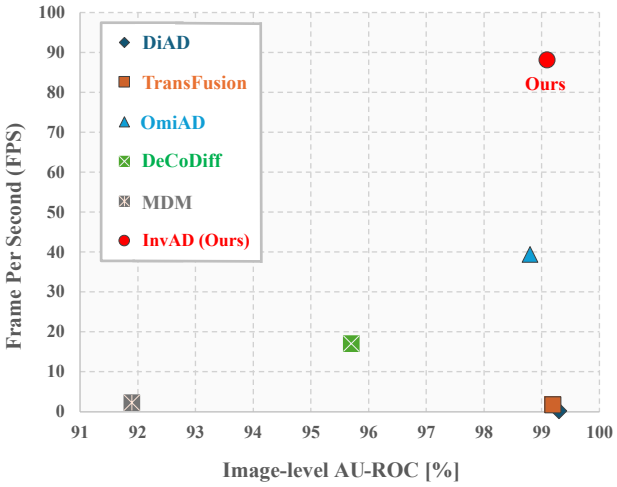


Figure 1. **Accuracy *v.s.* Speed** relationship of diffusion-based AD methods [5–9] on MVTecAD. *Our proposed* ***InvAD*** *achieves state-of-the-art AD performance with a substantial speedup.*

Due to the robust capabilities for approximating distributions, diffusion models [10–12] have emerged as powerful tools for AD by precisely modeling normal data distribution [5–8, 13–21]. The prevailing paradigm in these methods is to treat anomalies as noise: by adding noise of a certain intensity to an anomalous image and then denoising it with a diffusion model trained solely on normal data, the anomalous regions can be effectively reconstructed as normal regions. The anomaly score is therefore measured with the mean squared error (MSE) between the original input and its reconstructed counterpart, as shown in Figure 2 (a).

We argue that the above "*detection via denoising in RGB space*" paradigm is less effective given the following fundamental limitations: *(1) Noise-strength sensitiv-*

*Corresponding Author.

Table 1. **Comparison of the properties** of diffusion-based AD methods. Normal-only means whether the method involves pseudo-anomalies in training, NFE stands for the number of function evaluations, and TS refers to the timestep.

Method	Normal-only	NFE	FPS	TS Tuning-free	Multi-class	Scoring scheme
DiffAD *[ICCV'23]*	✓	N/A	N/A	✗	✗	Mask prediction
DiAD *[AAAI'24]*	✗	10	1.5	✓	✓	$\text{MSE}(g_\phi(\mathbf{x}_0), g_\phi(\hat{\mathbf{x}}_0))$
GLAD *[ECCV'24]*	✓	750	0.2	✓	✗	$\text{MSE}(g_\phi(\mathbf{x}_0), g_\phi(\hat{\mathbf{x}}_0))$
TransFusion *[ECCV'24]*	✓	20	1.6	✗	✗	Mask prediction
MDM *[ICML'25]*	✓	40	1.9	✗	✗	$\text{MSE}(g_\phi(\mathbf{x}_0), g_\phi(\hat{\mathbf{x}}_0))$
OmiAD *[ICML'25]*	✓	1	39.4	✗	✓	$\text{MSE}(g_\phi(\mathbf{x}_0), g_\phi(\hat{\mathbf{x}}_0))$
DeCo-Diff *[CVPR'25]*	✓	10	17.0	✗	✓	$\text{MSE}(g_\phi(\mathbf{x}_0), g_\phi(\hat{\mathbf{x}}_0))$
InvAD (Ours)	✓	**3**	**88.1**	✓	✓	$\log p_\theta(\mathbf{x}_T)$

ity. The detection performance depends heavily on the selection of appropriate noise strengths. Stronger noise unnecessarily corrupts normal regions, increasing false positives, while weaker noise leads to overly well-recovered anomaly regions [14, 15]; *(2) Computational expense in multi-step denoising*. Satisfactory reconstruction via diffusion models often requires iterative denoising, which is time-consuming for real-world applications. As listed in Table 1, most reconstruction-based methods involve high inference latency.

Therefore, we introduce the "*detection via noising in latent space*" paradigm, which bypasses the reconstruction process in prior AD approaches. Our core insight is that, since the learned diffusion model is solely aware of the normal data, we can utilize such learned normal data distribution through latent inversion, rather than reconstruction, as shown in Figure 2 (b). Specifically, for a given image, we directly infer the corresponding latent variable at the final diffusion step by tracing the probability flow ordinary differential equation (PF-ODE [22]) trajectory. Since PF-ODE is by nature deterministic, this constructs a one-to-one correspondence between the noise prior distribution and the distribution of normal images, ensuring that normal images can be ideally mapped to focus on a high-density region. Conversely, anomalous images will be mapped to low-density regions. Hence, we can detect anomalies by evaluating the degree of deviation of the inferred latent variable under the prior distribution.

While the above method detours noise strength tuning, it does not straightforwardly lead to a reduction of detection speed. In particular, the high inference latency primarily stems from the high-order ODE solver introduced for accurate reconstruction from inverted latents. To address this, we are inspired by [19, 23] to measure the norm of the score function as a proxy for typicality, which is robust to the step size selection of the Euler method. This allows us to enforce very few inversion steps to diffuse the clean image to pursue inference efficiency. Moreover, it contributes to a significantly more efficient inference paradigm compared to prior approaches. Importantly, because the learned diffusion model enables noise to be adaptively added during inversion, the key features of the original clean testing image can still be leveraged to harvest high detection accuracy. As plotted in Figure 1, our InvAD achieves state-of-the-art detection performance with the inference efficiency surpassing prior diffusion-based AD models by a large margin. Also, as our method is designed for the inference stage, it can be applied in a plug-and-play fashion to be incorporated into existing AD models. Experimental evaluations demonstrate that our method substantially improves inference latency, yet still yields high AD performance, across four widely-used AD benchmarks.

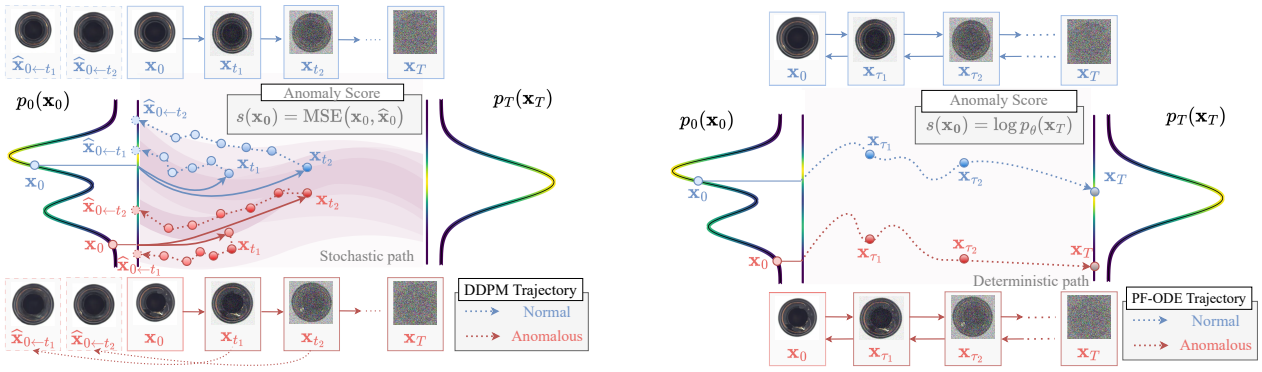
Our core contributions include three folds:

- Identifying the inherent limitations of existing reconstruction-based methods in achieving an optimal efficiency-accuracy balance, which we attribute fundamentally to the "detection via denoising in RGB space" paradigm.
- Proposing a novel paradigm, detection via noising in latent space, with an inversion-based AD framework **InvAD** that eliminates reconstruction requirements by directly estimating latent variables from inputs and quantifying anomaly scores through deviation measurements under the prior distribution.
- Demonstrating that our inversion-based approach, which infers latent variables, is simpler, faster, and achieves state-of-the-art detection performance compared to conventional reconstruction-based methods.

2. Background

2.1. Diffusion Models

Diffusion modeling [10, 22] is a recently emerged generative modeling framework, which aims to reverse a Markov



(a) Conventional reconstruction-based AD paradigm

(b) Our inversion-based AD paradigm

Figure 2. **Conceptual comparison** of conventional and our proposed AD paradigm. Conventional reconstruction-based paradigm (a) first perturbs an input sample $\mathbf{x}_0$ to a latent state $\mathbf{x}_t$ at step t , and then denoises $\mathbf{x}_t$ back to $\mathbf{x}_0$. The anomaly score is computed as the mean squared error (MSE) between the original input and its reconstructed sample. In contrast, our inversion-based paradigm (b) directly infers the latent state at the final step, $\mathbf{x}_T$, by tracing the PF-ODE trajectories. The anomaly score is then determined based on the typicality of $\mathbf{x}_T$ within the tractable latent distribution.

perturbation process (*forward* process). In general, each transition of the *forward* process is defined as:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}\Big(\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\,\mathbf{x}_{t-1}, \big(1 - \frac{\alpha_t}{\alpha_{t-1}}\big)\mathbf{I}\Big), \quad (1)$$

where the monotonically decreasing sequence $\alpha_{1:T} \in (0, 1]^T$ schedules the noise intensity. The sampling procedure can then be achieved with a *reverse* process, which defines a Markov process with a Gaussian transition kernel:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}\big(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta^{(t)}(\mathbf{x}_t), \big(1 - \frac{\alpha_t}{\alpha_{t-1}}\big)\mathbf{I}\big), \quad (2)$$

where θ denotes the learnable parameter of the DDPM. The training is realized by minimizing the ϵ-prediction loss:

$$\mathcal{L}_\gamma\left(\epsilon_\theta\right) := \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\left\| \epsilon_\theta^{(t)}(\mathbf{x}_t) - \epsilon_t \right\|_2^2 \right], \quad (3)$$

where $\mathbf{x}_t$ is efficiently sampled from $q_t(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1 - \alpha_t)\mathbf{I})$, and $\gamma := [\gamma_1, \ldots, \gamma_T]^T$ denotes the weighting vector. During sampling, one can first sample a pure noise from $p_\theta(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$, and then progressively trace Equation (2) for denoising.

2.2. Denoising Diffusion Implicit Models

The Gaussian approximation in Equation (2) generally works satisfactorily when the total number of diffusion steps, T, is sufficiently large [10]. However, increasing T also magnifies the number of function evaluations (NFE), which slows down the sampling procedure. To address this, Song *et al.* introduced denoising diffusion implicit models (DDIM), allowing for faster sampling by leveraging a subset of the original diffusion process [24]. Specifically, they reformulate the *reverse* process as follows:

$$\mathbf{x}_{\tau_{i-1}} = \sqrt{\alpha_{\tau_{i-1}}} f_\theta(\mathbf{x}_{\tau_i}) + \sqrt{1 - \alpha_{\tau_{i-1}}}\, \epsilon_\theta^{(\tau_i)}\big(\mathbf{x}_{\tau_i}\big), \quad (4)$$

where $f_\theta(\mathbf{x}_{\tau_i}) = \mathbf{x}_{\tau_i} - \sqrt{1 - \alpha_{\tau_i}}\, \epsilon_\theta^{(\tau_i)}\big(\mathbf{x}_{\tau_i}\big)/\sqrt{\alpha_{\tau_i}}$ and $\tau_S = [\tau_1, \tau_2, \ldots, \tau_S = T] \subset \{1, 2, \ldots, T\}$. This deterministic sampling equation can be interpreted as the Euler integration of the corresponding PF-ODE:

$$\mathrm{d}\mathbf{y}_t = \epsilon_\theta^{(t)}\, \mathrm{d}p_t, \quad (5)$$

where $\mathbf{y}_t := \mathbf{x}_t/\sqrt{\alpha_t}$ and $p_t := \sqrt{1/\alpha_t - 1}$. The PF-ODE gradually transforms a noise distribution into a data distribution, and vice versa. This approach, known as *inversion*, maps a given data sample back to its noise representation.

2.3. Diffusion-Based Anomaly Detection

The above diffusion-based generative models have been recently employed to handle AD. Existing diffusion-based AD methods generally follow the reconstruction-based paradigm. In particular, to address the challenge of noise level selection inherent in this framework, these approaches can be broadly classified into the following three categories:

- *Conditional information injection*-based methods incorporate auxiliary information, such as class labels [5], semantic features [5, 17, 21], or unmasked regions [7, 9, 25]. As pointed out in [17], conditioning the denosing process with such additional information contributes to reducing reconstruction error in the normal region and increasing it for anomalous regions. However, since this line of methods conditions the diffusion model at training time, they can lead to increased training complexity.

- *Discriminator*-based methods introduce an additional discriminator, typically trained to distinguish between normal and anomalous pixels [13, 16]. While this line of methods enables efficiency, they rely on pseudo-anomalies, biasing AD towards specific anomaly types.
- *Adaptive timesteps*-based methods attempt to adjust or identify perturbation timesteps by comparing the reconstruction errors of denoised results across different time steps [14]. As the above two types of methods do not incorporate an explicit mechanism for adjusting the perturbation time step, it is typically treated as a hyperparameter or addressed via ensembling over multiple time steps [15]. See the 5th column in Table 1 for more details.

In contrast to the state-of-the-art non-diffusion method [4] that achieves over 200 FPS, current diffusion-based methods mostly reach 1 FPS on average. While a recent diffusion-based approach [7] tackles this by introducing score distillation to constrain the diffusion iteration to only one step, the training for distillation can be complicated due to the instability of the adversarial learning. Table 1 summarizes the properties of current diffusion-based AD methods. Overall, a simple, efficient, yet effective paradigm for diffusion-based AD remains unexplored to date.

3. Methodology

To address the limitations of current diffusion-based methods, we introduce a novel diffusion-based AD framework with *latent inversion*, where the sampling trajectory is reformulated as $\mathbf{x}_0 \rightarrow \mathbf{x}_T$. In contrast to conventional reconstruction-based approaches $\left(\mathbf{x}_0 \rightarrow \mathbf{x}_t \rightarrow \hat{\mathbf{x}}_0\right)$, our method infers the final latent state $\mathbf{x}_T$ directly from $\mathbf{x}_0$ by following the **learned** PF-ODE trajectory. Figure 2(b) illustrates our reconstruction-free AD paradigm, which leverages latent inversion to achieve efficient yet accurate AD.

3.1. Inversion-Based Anomaly Detection

Given an image $\mathbf{x}$ and a score function $\epsilon_\theta^{(t)}$ trained using Equation (3), our goal is to infer the latent representation $\mathbf{x}_T$ corresponding to $\mathbf{x}$ by tracing the PF-ODE trajectory. In this work, we employ the PF-ODE derived from the DDIM framework [24], as shown in Equation (5). By applying an Euler approximation to Equation (5) and integrating forward, we obtain the following discrete update equation for latent inversion:

$$\mathbf{x}_{t+1} = \sqrt{\alpha_{t+1}} f_\theta(\mathbf{x}_t) + \sqrt{1-\alpha_{t+1}}\, \epsilon_\theta^{(t)}\left(\mathbf{x}_t\right). \quad (6)$$

However, directly deriving $\mathbf{x}_T$ with Equation (6) involves iterative computation of $\epsilon_\theta^{(t)}$ for T times, leading to slower inference. Here, we propose to only use the partial process of the original diffusion process in Equation (1). Given the discrete timestep set $\{1, 2, \ldots, T\}$, we prepare a monotonically increasing subset $\boldsymbol{\tau}_S = [\tau_1, \tau_2, \ldots, \tau_S = T] \subset \{1, 2, \ldots, T\}$, where $S \in \{1, 2, \ldots, T\}$ is the number of the partial process. The inversion modeling can thus be accelerated by simply noising the test image with fewer timesteps stored in $\boldsymbol{\tau}_S$, which is given by:

$$\mathbf{x}_{\tau_{i+1}} = \sqrt{\alpha_{\tau_{i+1}}} f_\theta(\mathbf{x}_{\tau_i}) + \sqrt{1-\alpha_{\tau_{i+1}}}\, \epsilon_\theta^{(\tau_i)}\left(\mathbf{x}_{\tau_i}\right). \quad (7)$$

Intuitively, Equation (7) leads to a considerable speed-up for the inference stage when $T \gg S$. Nevertheless, it also inevitably sacrifices approximation precision as the step interval for the Euler integration is extended. *Recall here that our goal is to resolve AD rather than pursue accurate pixel-wise reconstruction.* As such, we argue that high AD performance does not necessarily require reconstruction precision. It should be noticed that, even with limited inversion steps, the inversion mechanism in projecting an anomaly pixel can nonetheless yield sufficiently low typicality. In this regard, despite the poorly reconstructed samples, an anomaly pixel can still be expectedly mapped to a low-density region within the noise prior to achieve high detection performance. This insight suggests that our inversion-based AD method is free from the accuracy-efficiency trade-off in conventional reconstruction-based methods. We next need to find an ideal space that best suits our diffusion-based AD modeling. See *Supp. Mat.* for more detailed derivations of inversion.

3.2. Diffusion Modeling in Feature Space

Current image AD methods [3, 4, 17, 26] heavily rely on pre-trained networks, referred to as *backbones*, that are trained on large-scale datasets such as ImageNet [27]. In diffusion-based AD, these backbones are used to compare the original image with its reconstructed counterpart in feature space [5, 15, 17]. Because backbone features exhibit invariance to low-level variations (*e.g.*, noise or lighting conditions), they can significantly enhance AD performance.

However, in our inversion-based approach, the inferred latent $\mathbf{x}_T$ follows a standard Gaussian distribution, not the data distribution. To nonetheless utilize backbone models, we design our method to follow the latent-diffusion manner by extracting the feature $\mathbf{z}$ via $\mathbf{z} = g_\phi(\mathbf{x}) \in \mathbb{R}^{C \times h \times w}$, where $g_\phi(\cdot)$ denotes the backbone model, and $C, (h, w)$ mean the feature map dimension and resolutions, respectively. This feature-space diffusion modeling enables our inference stage to refer to high-level semantics, rather than low-level pixel patterns. Also, such a latent-diffusion-based framework leads to further inference efficiency.

3.3. Anomaly Score Calculation

Once we obtain the latent representation $\mathbf{z}_T \in \mathbb{R}^{C \times h \times w}$ for features of a given test sample $\mathbf{x}$, the most straightforward way to compute the anomaly score is via the log-likelihood under the latent Gaussian distribution. However, it often

Algorithm 1 Inference with InvAD

Ensure: $\epsilon_\theta^{(t)}$, g_ϕ, $\mathbf{x}_0 \in \mathbb{R}^{3\times H\times W}$, $\tau_S = [\tau_1, \ldots, \tau_S]$, $\alpha_{1:T}$
Require: Anomaly score s, Anomaly map $A \in \mathbb{R}^{H\times W}$

```
/* DDIM Inversion */
Initialize z ← g_φ(x_0) ∈ R^{C×h×w}
for i = 0 to S − 1 do
    Update z with Equation 7
    if i = S − 1 then
        break
    end if
end for
/* Anomaly scoring */
Initialize A ← 0 ∈ R^{h×w}
for u ∈ {0, ..., h − 1}, v ∈ {0, ..., w − 1} do
    Update A[u, v] with Equation 8
end for
Upsample A ∈ R^{h×w} to R^{H×W}
s ← max(A) − min(A) + Σ_{u,v} A[u, v]
return (s, A)
```

triggers the known *reverse-scoring* problem and fails to capture small anomalies (See *Supp. Mat.*). Given that anomalies are inherently sparse in images, we additionally propose a norm-based image-level anomaly scoring approach. Specifically, we compute the Euclidean norm over all channels C for each pixel (i, j) in $\mathbf{z}_T$, which is written as:

$$\mathbf{z}_T^{\text{normed}}[i, j] = \|\mathbf{z}_T[:, i, j]\|_2 \quad . \tag{8}$$

Different from [5], for the image-level anomaly score, we compute the difference between the maximum and minimum values of $\mathbf{z}_T^{\text{normed}}$. This is motivated by the observation that, even if an image is anomalous, anomalies are often locally and sparsely distributed within small regions, and the majority of image regions remain normal. In essence, our scheme for image-level anomaly scoring examines the difference between the highest- and the lowest-norm pixels to mitigate the influence of outliers, thereby alleviating the reverse-scoring issue. The pixel-level anomaly score A is simply produced by extending $\mathbf{z}_T^{\text{normed}} \in \mathbb{R}^{h\times w}$ to match the desired output resolution (H, W) via bilinear interpolation.

Our proposed InvAD paradigm is summarized in Algorithm 1. It is efficient yet effective, and can be applied in a plug-and-play manner for existing diffusion-based AD models, as it is devised only for the inference stage.

4. Experiments

4.1. Datasets

We use three widely-used unsupervised AD benchmarks—MVTecAD [28], VisA [29], MPDD [30]—all designed for measuring real-world AD performance on RGB industrial images. We also experiment on the BMAD [31] benchmark (with six reorganized datasets) to study the effectiveness of InvAD on the medical AD domain. Since each benchmark provides both image- and pixel-level annotations, we can evaluate performance in terms of overall AD (image-level) as well as precise localization (pixel-level). See *Supp. Mat.* for a detailed description of these datasets.

4.2. Evaluation Metrics

Following [7], we employ Area Under the Receiver Operating Characteristic Curve (AU-ROC), Average Precision (AP), F1-score max (F1_max) for both image- and pixel-level AD evaluation. We also adopt the Area Under the Per-Region-Overlap (AU-PRO) [28] metric to examine pixel-level anomaly localization. The Frames Per Second (FPS) metric is further applied to study the efficiency of inference.

4.3. Implementations

To assess the effectiveness of the proposed method, we begin by training a Diffusion Transformer (DiT)-based diffusion model [32] from scratch on each dataset with Equation (3), where we set $\gamma = \mathbf{1}$ as DDPM [10]. To ensure fair comparisons, we employ the pre-trained EfficientNet-B4 [33] model on ImageNet-1k [27] to encode the images into feature space as OmiAD [7] and HVQ-Trans [34]. We train the diffusion model for 300 epochs across all datasets using the AdamW optimizer [35]. The total diffusion step T is set to 1000 during training using a linear noise schedule. For all experiments, we uniformly set $S = 3$ to skip intermediate steps within the original T steps, and create the subset $\tau_3 = [333, 666, 999]$ to acclerate inversion compuatation. We conduct all throughput measurements on an NVIDIA RTX 4090 (24GB) with batch size set to 64. See *Supp. Mat.* for more details.

4.4. Comparison with SOTA AD Methods

We compare the quantitative results against prior multi-class AD competitors, including *(i) Transformer-based methods* (HVQ-Trans [34], UniAD [36]), *(ii) reconstruction-based methods* (RD4AD [37], DeSTSeg [38], SimpleNet [39]), *(iii) diffusion-based methods* (DiAD [5], OmiAD [7], MDM [7]). The results are summarized in Table 2. It can be seen that our method generally yields the best image-level and highly competitive pixel-level detection performance on three datasets. Notably, our method consistently outperforms all the compared methods regarding inference efficiency by a large margin (*i.e.*, FPS: 88.1 *v.s.* 39.4 on MVTecAD). This can be attributed to the property of our proposed inversion-based AD paradigm, which enables high detection accuracy with fewer NFEs for acceleration. By contrast, conventional diffusion-based methods, such as DiAD [5], require VAE decoding for reconstruction in the pixel space, thus resulting in slower

Table 2. **Quantitative results** on different AD datasets under the multi-class setting. The best and the second-best results are highlighted in bold and underlined, respectively. For each row, the mAD averages the evaluation over all image- and pixel-level metrics (*i.e.*, the 3rd to the 9th column). FPS measures the detection efficiency.

Dataset	Method	Image-level			Pixel-level				mAD	FPS
		AU-ROC	AP	F1_max	AU-ROC	AP	F1_max	AU-PRO		
MVTecAD	RD4AD *[CVPR'22]*	94.6	96.5	95.2	96.1	48.6	53.8	91.1	82.3	4.8
	UniAD *[NeurIPS'22]*	96.5	98.8	96.2	96.8	43.4	49.5	90.7	81.7	5.3
	SimpleNet *[CVPR'23]*	95.3	98.4	95.8	96.9	45.9	49.7	86.5	81.2	N/A
	DeSTSeg *[CVPR'23]*	89.2	95.5	91.6	93.1	**54.3**	50.9	64.8	77.1	N/A
	DiAD *[AAAI'24]*	97.2	99.0	96.5	96.8	52.6	55.5	90.7	84.0	0.1
	HVQ-Trans *[NeurIPS'23]*	98.0	99.5	97.5	97.3	48.2	53.3	91.4	83.6	5.6
	MDM *[ICML'25]*	87.2	94.2	91.0	94.8	39.5	44.8	88.0	77.1	1.9
	OmiAD *[ICML'25]*	98.8	**99.7**	**98.5**	**97.7**	52.6	**56.7**	**93.2**	**85.3**	39.4
	InvAD (Ours)	**99.0**	99.6	**98.5**	97.5	46.5	52.3	92.7	83.7	**88.1**
VisA	RD4AD *[CVPR'22]*	92.4	92.4	89.6	98.1	38.0	42.6	91.8	77.8	4.9
	UniAD *[NeurIPS'22]*	88.8	90.8	85.8	98.3	33.7	39.0	85.5	74.6	4.6
	SimpleNet *[CVPR'23]*	87.2	87.0	81.8	96.8	34.7	37.8	81.4	72.4	N/A
	DeSTSeg *[CVPR'23]*	88.9	89.0	85.2	96.1	39.6	43.4	67.4	72.8	N/A
	DiAD *[AAAI'24]*	86.8	88.3	85.1	96.0	26.1	33.0	75.2	70.1	0.1
	HVQ-Trans *[NeurIPS'23]*	93.2	92.8	87.6	98.7	35.0	39.6	86.3	76.2	5.0
	OmiAD *[ICML'25]*	95.3	96.0	91.2	98.9	**40.4**	**44.1**	89.2	79.3	35.3
	InvAD (Ours)	**96.9**	**97.2**	**93.7**	**99.1**	39.2	43.1	**92.9**	**80.3**	**74.1**
MPDD	RD4AD *[CVPR'22]*	84.1	83.2	84.1	98.1	35.2	38.7	93.4	73.8	4.7
	UniAD *[NeurIPS'22]*	82.2	87.1	85.1	95.1	18.9	25.0	81.9	67.9	5.8
	SimpleNet *[CVPR'23]*	90.6	94.1	89.7	97.1	33.6	35.7	90.0	75.8	N/A
	DeSTSeg *[CVPR'23]*	93.0	95.1	90.6	94.1	33.2	37.6	59.8	71.9	N/A
	DiAD *[AAAI'24]*	74.6	82.1	82.5	93.0	15.9	21.2	78.4	64.0	0.1
	HVQ-Trans *[NeurIPS'23]*	86.5	88.1	85.8	96.7	27.6	31.4	86.9	71.9	6.2
	OmiAD *[ICML'25]*	93.7	95.5	90.9	**98.6**	37.6	42.3	**94.0**	78.9	49.8
	InvAD (Ours)	**96.5**	**96.5**	**94.4**	98.3	**40.1**	**43.9**	91.2	**80.1**	**120**

Table 3. **Quantitative results** on BMAD under the single-class setting, where we report image-level AU-ROC (det.), pixel-level AU-ROC (loc.), and FPS. The best and the second-best results are highlighted in bold and underlined, respectively. For each row, the mAD averages the evaluation over all image- and pixel-level metrics. *FPS measures the detection efficiency on the RESC dataset.*

Methods	BraTS2021		BTCV + LiTS		RESC		OCT2017	RSNA	Camelyon16	mAD	FPS
	det.	loc.	det.	loc.	det.	loc.	det.	det.	det.		
PaDiM *[ICPR'20]*	79.0	94.4	50.8	90.9	75.9	91.4	91.8	77.5	67.3	79.9	20
CFlow *[CVPR'20]*	74.8	93.8	50.8	92.4	75.0	93.8	85.4	71.5	55.7	77.0	15
RD4AD *[CVPR'22]*	89.5	96.5	60.4	96.0	87.8	96.2	97.3	67.6	66.8	84.2	20
PatchCore *[CVPR'22]*	**91.7**	97.0	60.3	96.4	91.6	**96.5**	98.6	76.1	**69.3**	86.4	20
SimpleNet *[CVPR'23]*	82.5	94.8	**72.3**	97.5	76.2	77.1	94.7	69.1	62.4	80.7	10
InvAD (Ours)	90.3	**97.8**	60.6	**97.8**	**93.6**	**96.5**	**99.0**	**82.7**	67.7	**87.2**	**88**

inference. It is noteworthy that our method is also faster than OmiAD whose NFE is 1, which is primarily due to the heavy diffusion backbone and its modeling strategy in high-resolution feature space.

The results against diffusion-based AD models on single-class MVTecAD are displayed in Figure 1. Similarly to the multi-class setting, InvAD yields substantially reduced inference latency with state-of-the-art performance, indicating the effectiveness of inversion modeling for AD.

We next evaluate on BMAD to study the effectiveness of our InvAD on the medical domain. Here, we compare against PaDiM [40], CFlow [41], RD4AD [37], PatchCore

Table 4. **Comparison of reconstruction-based approach (Recon.) against our inversion-based AD paradigm**, under different total diffusion steps S and perturbation ratios r, in multi-class MVTecAD with image-level AU-ROC.

	Ratio r	Total Diffusion Steps S					
		3	**5**	**10**	**50**	**100**	**1000**
Recon.	10 %	N/A	N/A	66.6	97.7	97.9	98.0
	20 %	N/A	64.7	97.9	91.3	98.2	98.2
	40 %	64.9	75.0	89.4	98.0	98.2	98.2
	60 %	64.9	68.4	87.1	97.3	97.8	98.1
	80 %	67.8	67.8	74.1	91.8	92.1	93.9
Inversion (Ours)		**99.0**	98.9	98.4	96.0	95.7	95.4

Table 5. **Generalizability evaluation** of our method when incorporated into other diffusion-based AD approaches on MVTecAD with image- and pixel-level AU-ROC, number of total parameters (#Params) and FPS.

Method	**MVTecAD**		**FPS**
	det.	loc.	
DiAD *[AAAI'24]*	97.2	96.8	0.1
DiAD + **InvAD**	**98.2** (+1.0)	**97.5**(+0.7)	**88.1**(+88)
MDM *[ICML'25]*	91.9	94.8	2.2
MDM + **InvAD**	**98.2** (+6.3)	**97.5** (+2.7)	**63** (+60.8)

[3], and SimpleNet [39]. We report image-level AU-ROC (det.), and pixel-level AU-ROC (loc.), if the ground-truth anomaly maps are provided. The results are presented in Table 3. It can be confirmed that InvAD achieves state-of-the-art performance across diverse BMAD medical datasets with a substantially improved inference efficiency. This further evidences the capacity and robustness of our method in handling medical AD, other than industrial inspection.

4.5. Is Inversion Effective?

Our approach builds on the inversion modeling to realize reconstruction-free AD. To investigate the validity of this design, we vary two key parameters in reconstruction-based methods, total diffusion steps S and perturbation timesteps t, to gain a deeper insight into the effect. In particular, instead of inspecting t directly, we study the ratio of $r = t/S$ as it better captures the perturbation strength across different S. The results are summarized in Table 4, where we only modify the inference stage to examine the accuracy (*i.e.,* image-level AU-ROC) change. It can be confirmed that for reconstruction-based methods, the parameter tuning efforts can be highly burdensome to yield ideal performance. While a larger S tends to perform better, it induces computationally expensive denoising iterations.

By contrast, our inversion-based method can achieve the best performance even with an extremely small inversion step setting (*e.g.*, $S = 3, 5$ in the bottom row of Table 4), with no reliance on perturbation timesteps as it is not involved in our modeling. This further evidences the superiority of our InvAD for the task of image AD.

Table 6. **Component ablation** on multi-class MVTecAD. FDM denotes feature space diffusion models. S-Inv and M-Inv denote single- and multi-step inversion, resepectively.

Setting	**FDM**	**S-Inv**	**M-Inv**	**mAD**
A1	✓			57.3
A2		✓		44.9
A3	✓	✓		71.0
A4 (**Ours**)	✓		✓	**83.7**

4.6. Generalizability Evaluation

As our InvAD is designed for the inference stage, it can be incorporated into existing diffusion-based AD models in a plug-and-play manner. To evaluate the effectiveness, we incorporate InvAD into the inference stage for DiAD [5] and MDM [7], and report the image- and pixel-level AU-ROC and the FPS metrics on multi-class MVTecAD. It can be seen from Table 5 that InvAD leads to an accuracy gain for both methods with substantial inference acceleration. In particular, since MDM learns to optimize the noise residual during training, it suits better for our method, as reflected in the remarkable improvement of accuracy. This suggests that InvAD is model-agnostic and can be inserted as a plug-and-play module into different diffusion-based AD techniques.

4.7. Component Ablation

Our framework consists of two components: (i) feature space diffusion model (FDM), and (ii) multi-step inversion. To individually validate the contribution, we evaluate the AD performance of each component as well as their combinations on the multi-class MVTecAD. Table 6 reports the results of four scenarios (A1-A4). In the case of A1, we skip the inversion (lines 3-8 of Algorithm 1), then calculate the anomaly score as in [42]. This leads to a significantly degredation in AD performance compared with our complete framework (A4), highlighting the importance of inversion. Furthermore, for the FDM-free case of A2 (*i.e.,* pixel-space diffusion), single-step inversion also degrades accuracy. Combining FDM and single-step inversion (A3) substantially improves AD accuracy, and multi-step inversion (A4) brings further accuracy gains.

4.8. Backbone Design Ablation

Our inversion-based method is considered in the pre-trained feature space rather than pixel space. In general, the selection of feature extractors heavily impacts the performance of AD methods [5, 17, 46]. In Table 7, we evaluate

Table 7. **Design ablation of feature space encoding modules** on multi-class MVTecAD with image- and pixel-level AU-ROC, number of total parameters (#Params), and FPS.

Model	MVTecAD		# Params	FPS
	det.	loc.		
ViT-B [43]	94.2	95.5	1312M	85.3
DINO-B [44]	96.1	**97.7**	1312M	85.3
EfficientNet-B4 [33]	**99.0**	96.9	**1243M**	**88.1**

Table 8. **Design ablation of diffusion architectures** on multi-class MVTecAD with image- and pixel-level AU-ROC, and FPS.

Model	MVTecAD		# Params	FPS
	det.	loc.		
MLP [45]	97.2	96.8	351M	241
UNet [10]	98.0	97.3	1433M	**288**
DiT-base [32]	93.8	95.9	**155M**	240
DiT-gigant [32]	**99.0**	**97.5**	1223M	88

Table 9. **Scoring scheme ablation** under different total diffusion steps S in multi-class MVTecAD with image-level AU-ROC.

Total Diffusion Steps S	3	5	10	50	100	1000
NLL	96.1	95.2	93.0	89.7	89.4	89.1
Diff	98.4	98.1	95.8	83.2	82.2	82.0
NLL + Diff (Ours)	**99.0**	**98.9**	**98.4**	**96.0**	**95.7**	**95.4**

the performance with different backbones including ViT-B [43], DINO-B [44], and EfficientNet-B4 [33]. As expected, stronger backbones lead to better performance. It should be noted that EfficientNet is a common backbone in diffusion-based AD [7, 8, 21]. We follow this standard choice for fair comparisons, while contributing a novel inversion-based paradigm to jointly harvest accuracy and efficiency.

Also, we study the influence of the design of diffusion modeling backbones on AD performance in Table 8. In particular, we compare four types of backbones: MLP, UNet, DiT-base, and DiT-gigant [32]. In contrast to the feature encoding modules, we observe that the performance is generally similar, even with the simple MLP implementation. This suggests that our method is robust to diffusion architectures, but also relies on the quality of embeddings derived with feature extractors, as prior methods [3, 7]. See *Supp. Mat.* for a detailed description of backbones.

4.9. Scoring Scheme Ablation

As explained in Section 3.3, our anomaly scoring scheme considers both negative-log-likelihood (NLL) in the noise prior and the spatial difference of Euclidean norm (Diff) to alleviate the *reverse-scoring* problem [47, 48]. To study the advantage of this strategy, we compare the performance in Table 9 with the conventional NLL-based scoring on MVTecAD. It can be seen that our scoring scheme (*i.e.*, NLL + Diff) contributes to robustness to the diffusion step setting S, while NLL or Diff solely do not achieve this. This demonstrates that our scoring scheme plays a key role in the proposed InvAD paradigm to realize reconstruction-free AD, where S can be set to be very small for efficiency.

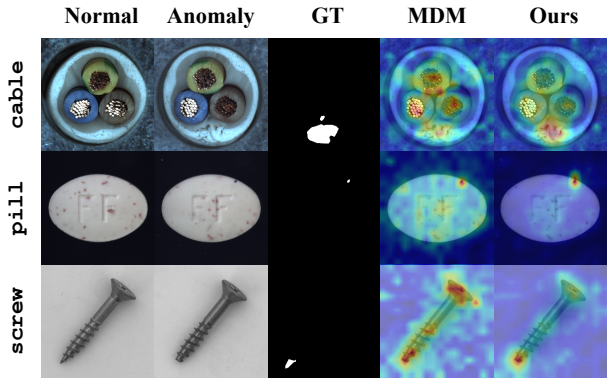


Figure 3. **Visualization of anomaly localization** against MDM [7] on MVTecAD. GT displays the ground-truth anomaly map.

4.10. Visualization Analysis

We qualitatively evaluate in Figure 3 the pixel-level AD performance of InvAD against MDM [7]. Similar to the quantitative evaluation in Table 2, InvAD generally achieves more precise anomaly localization with fewer false positives than MDM, which again, demonstrates that our method improves efficiency and accuracy while preserving interpretability for anomaly locations without reconstruction.

5. Conclusion

We propose a simple yet effective diffusion-based AD paradigm, "*detection via noising in latent space*", to address the fundamental challenges within the conventional "*detection via denoising in RGB space*". Our method, **InvAD**, is reconstruction-free and directly infers the final latent state with the DDIM inversion mechanism. To pursue efficiency, we simply enforce very few inversion steps to noise a test sample, yet manage to maintain high detection accuracy as the typicality can still be accurately scored. Due to the reconstruction-free design, InvAD is inherently tuning-free for the perturbation timestep. We experimentally demonstrate that InvAD achieves state-of-the-art detection performance, especially regarding inference efficiency.

Limitation and Future Work. While our method can achieve satisfactory performance, it still needs more than one NFE. To further improve efficiency (*i.e.*, 1 NFE), we will consider introducing diffusion distillation to compactly compress the progressive inversion process into only one single step. Also, exploring improved or task-specific inversion mechanism is an interesting future direction.

6. Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP25H01110 and the TAKEUCHI research grant.

References

[1] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang. Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges. *Materials (Basel)*, 13(24):5755, December 2020. 1

[2] Cosmin I. Bercea, Benedikt Wiestler, Daniel Rueckert, and Julia A. Schnabel. Generalizing unsupervised anomaly detection: Towards unbiased pathology screening. In *MIDL*, 2023. 1

[3] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *CVPR*, 2021. 1, 4, 7, 8

[4] Kilian Batzner, Lars Heckler, and Rebecca König. EfficientAD: Accurate visual anomaly detection at millisecond-level latencies. In *WACV*, 2023. 1, 4

[5] Haoyang He, Jiangning Zhang, Hongxu Chen, Xuhai Chen, Zhishan Li, Xu Chen, Yabiao Wang, Chengjie Wang, and Lei Xie. A diffusion-based framework for multi-class anomaly detection. In *AAAI*, pages 8472–8480, March 2024. 1, 3, 4, 5, 7, 2

[6] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. TransFusion: A transparency-based diffusion model for anomaly detection. In *ECCV*, pages 91–108, Cham, 2024. Springer Nature Switzerland.

[7] Yaoxuan Feng, Wenchao Chen, Yuxin Li, Bo Chen, Yubiao Wang, Zixuan Zhao, Hongwei Liu, and Mingyuan Zhou. OmiAD: One-step adaptive masked diffusion model for multi-class anomaly detection via adversarial distillation. In *ICML*, 2025. To appear. 3, 4, 5, 7, 8, 1

[8] Farzad Beizaee, Gregory A. Lodygensky, Christian Desrosiers, and Jose Dolz. Correcting deviations from normality: A reformulated diffusion model for multi-class unsupervised anomaly detection. In *CVPR*, pages 19088–19097, 2025. 1, 8

[9] Farzad Beizaee, Gregory Lodygensky, Christian Desrosiers, and Jose Dolz. MAD-AD: Masked diffusion for unsupervised brain anomaly detection. In *IPMI*, pages 139–153. Springer, 2025. 1, 3

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 2020. 1, 2, 3, 5, 8

[11] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

[12] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. 2021. 1

[13] Xinyi Zhang, Naiqi Li, Jiawei Li, Tao Dai, Yong Jiang, and Shu-Tao Xia. Unsupervised surface anomaly detection with diffusion probabilistic model. In *ICCV*, pages 6759–6768, 2023. 1, 4

[14] Hang Yao, Ming Liu, Haolin Wang, Zhicun Yin, Zifei Yan, Xiaopeng Hong, and Wangmeng Zuo. Glad: Towards better reconstruction with global and local adaptive diffusion models for unsupervised anomaly detection. *arXiv preprint arXiv:2406.07487*, 2024. 2, 4

[15] Fanbin Lu, Xufeng Yao, Chi-Wing Fu, and Jiaya Jia. Removing anomalies as noises for industrial defect localization. In *ICCV*, pages 16120–16129, October 2023. 2, 4

[16] Hui Zhang, Zheng Wang, Zuxuan Wu, and Yu-Gang Jiang. Diffusionad: Norm-guided one-step denoising diffusion for anomaly detection. *arXiv preprint arXiv:2303.08730*, 2023. 4

[17] Arian Mousakhan, Thomas Brox, and Jawad Tayyub. Anomaly detection with conditioned denoising diffusion models. *arXiv preprint arXiv:2305.15956*, 2023. 3, 4, 7

[18] Julian Wyatt, Adam Leach, Sebastian M. Schmon, and Chris G. Willcocks. AnoDDPM: Anomaly detection with denoising diffusion probabilistic models using simplex noise. In *CVPRW*, pages 649–655, 2022.

[19] Yapeng Teng, Haoyang Li, Fuzhen Cai, Ming Shao, and Siyu Xia. Unsupervised visual anomaly detection with score-based generative model, 2023. 2

[20] Victor Livernoche, Vineet Jain, Yashar Hezaveh, and Siamak Ravanbakhsh. On diffusion modeling for anomaly detection. In *ICLR*, 2024.

[21] Yuxin Li, Yaoxuan Feng, Bo Chen, Wenchao Chen, Yubiao Wang, Xinyue Hu, Baolin Sun, Chunhui Qu, and Mingyuan Zhou. Vague prototype-oriented diffusion model for multi-class anomaly detection. In *ICML*, 2024. 1, 3, 8

[22] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 2

[23] Alvin Heng, Alexandre H. Thiery, and Harold Soh. Out-of-distribution detection with a single unconditional diffusion model. volume 37, pages 43952–43974, 2024. 2

[24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3, 4

[25] Rui Xu, Yunke Wang, and Bo Du. MAEDiff: Masked autoencoder-enhanced diffusion models for unsupervised anomaly detection in brain images. *arXiv preprint arXiv:2401.10561*, 2024. 3

[26] Taegeon Um, Byungsoo Oh, Byeongchan Seo, Minhyeok Kweun, Goeun Kim, and Woo-Yeon Lee. FastFlow: Accelerating deep learning model training with smart offloading of input data pipeline. *PVLDB*, 16(5):1086–1099, January 2023. 4

[27] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 4, 5

[28] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD: A comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, pages 9584–9592, 2019. 5, 1, 2

[29] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-Difference self-supervised pre-training for anomaly detection and segmentation. In *ECCV*, 2022. 5, 1

[30] Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. Deep learning-based defect detection of metal parts: Evaluating current methods in complex conditions. In *ICUMT*, pages 66–71, 2021. 5, 1

[31] Jinan Bao, Hanshi Sun, Hanqiu Deng, Yinsheng He, Zhaoxiang Zhang, and Xingyu Li. BMAD: Benchmarks for medical anomaly detection. In *CVPRW*, pages 4042–4053, 2024. 5, 1

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4172–4182, 2023. 5, 8, 1

[33] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 5, 8, 1

[34] Ruiying Lu, YuJie Wu, Long Tian, Dongsheng Wang, Bo Chen, Xiyang Liu, and Ruimin Hu. Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection. pages 8487–8500, December 2023. 5, 1

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 5

[36] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. volume 35, 2022. 5

[37] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *CVPR*, pages 9727–9736, 2022. 5, 6

[38] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. DeSTSeg: Segmentation guided denoising student-teacher for anomaly detection. In *CVPR*, pages 3914–3923, 2023. 5

[39] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. SimpleNet: A simple network for image anomaly detection and localization. In *CVPR*, pages 20402–20411, 2023. 5, 7

[40] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDiM: A patch distribution modeling framework for anomaly detection and localization. In *Pattern Recognition. ICPR International Workshops and Challenges*, pages 475–489, Cham, 2021. Springer International Publishing. 6

[41] Albert Pumarola, Stefan Popov, Francesc Moreno-Noguer, and Vittorio Ferrari. C-Flow: Conditional generative flow models for images and 3D point clouds. In *CVPR*, pages 7946–7955, 2020. 6

[42] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *ICPR*, pages 6726–6733, 2021. 7

[43] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 8, 1

[44] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9630–9640, 2021. 8, 1

[45] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. 2024. 8, 1

[46] Xincheng Yao, Yan Luo, Zefeng Qian, and Chongyang Zhang. ADPretrain: Advancing industrial anomaly detection via anomaly representation pretraining. 2025. 7

[47] Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *ICLR*, 2019. 8, 4

[48] Genki Osada, Takahashi Tsubasa, Budrul Ahsan, and Takashi Nishide. Out-of-distribution detection with reconstruction error and typicality-based penalty. In *WACV*, January 2023. 8, 4

[49] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. pages 8024–8035, 2019. 2

InvAD: Inversion-based Reconstruction-Free Anomaly Detection with Diffusion Models

Supplementary Material

This document supplements our main paper, "InvAD: Inversion-based Reconstruction-Free Anomaly Detection with Diffusion Models". The reference indices are consistent with those used in the main paper.

7. Dataset Details

We utilize four publicly available anomaly detection (AD) benchmarks for evaluation: MVTecAD [28], VisA [29], MPDD [30], and BMAD [31]. Table 10 summarizes the key statistics of the datasets used in our experiments, including the number of classes, domain, image resolution, and sample counts for training and testing. All of these datasets follow an unsupervised AD setting. In other words, the training set only consists of defect-free normal images, and the test set consists of both normal and anomalous images.

8. Implementation Details

Our implementation is available at: https://github.com/SkyShunsuke/InversionAD.

8.1. Data Preparation

We resize all training images to 256×256 pixels and do not apply any data augmentation. Image normalization is performed using the ImageNet dataset's channel-wise mean ([0.485, 0.456, 0.406]) and standard deviation ([0.229, 0.224, 0.225]).

8.2. Model Setup

We adopt the original DiT architecture [32] as the backbone of our diffusion modeling, but remove all class-conditioning so that our method does not depend on class labels. This design allows DiT to be applied in scenarios where ground-truth annotations are unavailable or impractical (*e.g.*, inspection of diverse products on a manufacturing line). We set the number of transformer blocks and embedding dimensions to 16 and 2048 for DiT-gigant, and to 8 and 1024 for DiT-base, respectively. For the UNet-based diffusion models evaluated in Table 8 (in the main paper), we follow the architectural specifications of the original DDPM implementation [10], consistent with prior work [5, 8]. For the MLP-based diffusion models, we stack multiple MLP blocks and apply AdaLN-Zero conditioning [32], as in [45]. We summarize all major hyperparameters for diffusion modeling in Table 11.

Table 10. **Dataset statistics** used in our experiments. 'Res.' represents image resolution, where $H = W$. 'N' and 'A' denote the numbers of normal and anomalous samples, respectively.

	MVTecAD	**VisA**	**MPDD**	**BMAD**
Classes	15	12	6	6
Domain	Industrial	Industrial	Metal	Medical
Res.	1024	256	1024	1024
Train (N)	2,629	8,659	888	52742
Test (N)	467	962	176	4540
Test (A)	1,258	1,200	282	22632

8.3. Backbone Setup

For fair comparisons with previous works [7, 34], we employ pre-trained EfficientNet-B4 [33] as our feature extractor. To capture anomalies at different scales, we first extract feature maps from the 1st to the 4th blocks from EfficientNet-B4, and then apply bilinear interpolation followed by channel-wise concatenation, creating $\mathbf{z} \in \mathbb{R}^{272\times16\times16}$, as in [7, 34].

We evaluate in Table 7 in the main paper the robustness of our method to the different feature space encoding modules: ViT-B [43] and DINO-base [44]. Both modules employ the vision transformer-based architecture and use ImageNet-1k as the pre-training dataset. The only difference comes from the pre-training scheme (*i.e.*, supervised for ViT-B [43], and self-distillation for DINO-base [44]). For both models, we extract patch tokens of the last block and reorder them to construct feature maps, creating $\mathbf{z} \in \mathbb{R}^{768\times16\times16}$.

8.4. Evaluation Pipeline

We evaluate AD performance at both image- and pixel-levels using the following metrics:

- **Area Under the Receiver Operating Characteristic Curve (AU-ROC)** provides an intuition on how well the model takes a trade-off between recall and precision in a threshold-independent manner.
- **Average Precision (AP)** conducts a more practical evaluation than AU-ROC, where the test set contains far fewer anomalous samples compared to normal ones.
- **Maximum F1-score ($\mathbf{F1_{max}}$)** measures the F1 score on the best-performed threshold, which reflects actual performance on the real problem.

To assess pixel-level anomaly localization, we employ:

- **Area Under the Per-Region-Overlap Curve (AU-PRO)**

Table 11. **Hyperparameters** of diffusion modeling.

Parameter	Value
Diffusion Settings	
Training diffusion steps (T)	1000
Inference diffusion steps (S)	3
Noise scheduler	Linear, $\beta_1 = 1 \times 10^{-4}$, $\beta_T = 0.02$
Parameterization	ϵ-prediction
Noise scale (σ)	Fixed
Training Settings	
Batch size	8
Optimizer	AdamW
Gradient clipping threshold	1.0
Initial learning rate	1×10^{-6}
Peak learning rate	5×10^{-5}
Final learning rate	5×10^{-6}
Learning rate scheduler	Warmup + cosine decay
Warmup epochs	40
Total number of epochs	300
Weight decay	0

Table 12. **Computing environment** in our experiments.

Component	Training	Evaluation
CPU	Xeon 8468 ×2	i9-14900KF
GPU	NVIDIA H200 ×8	RTX 4090 (24GB)
RAM	1.0 TiB (≈1024GB)	31 GB
OS	RHEL 9.4 (Plow)	Ubuntu 24.04.2 LTS

calculates the IoU between prediction and anomalous region, which is more robust to the small anomalies [28].

8.5. Experimental Environment

Our implementations are based on Pytorch-2.7.1 [49] and Python-3.10. We show the machine specification for the training and evaluation in our experiments in Table 12.

To ensure reproducibility, we fixed random seeds for all libraries (NumPy, PyTorch, etc.) and disabled nondeterministic GPU operations if possible.

It is important to understand that we do not train any extra conditioning modules nor perform reconstruction, just train unconditional diffusion models in the standard way. This leads to significant training time reduction (e.g., for MVTecAD training time, InvAD: 2.5h on 8×H200 @28 GB/GPU— 2 × faster than DiAD [5]'s 5h). The use of 8× H200 is optional, *not required*. Specifically, training can be performed on GPUs with more than 32 GB of memory for our largest models.

Table 13. **Inversion schedule ablation** under different total diffusion steps S in multi-class MVTecAD with mAD.

$g(u)$, $u = t/T$	$S = 3$	$S = 10$	$S = 100$
Uniform $g(u) = u$	**83.35**	**82.11**	**79.35**
Quad $g(u) = u^2$	82.37	80.76	79.14
Cube $g(u) = u^3$	81.68	80.84	79.18
Exp $g(u) = \frac{e^{5u}-1}{e^5-1}$	81.48	80.85	79.23

Table 14. **Feature resolution ablation** under different anomaly sizes in multi-class MVTecAD with mAD.

$h \times w$	**Tiny**	**Small**	**Medium**	**Large**	**All**
8×8	73.38	76.38	79.17	83.57	82.05
16×16	**78.54**	**80.65**	**81.76**	**84.94**	**83.70**
24×24	76.33	78.76	80.00	81.97	81.44

9. Addtional ablations

Inversion schedule. To accelerate inversion computation, we uniformly set $S = 3$ to skip intermediate steps within the original $T = 1000$ steps, and create the subset of timesteps $\tau_3 = [333, 666, 999]$ (*i.e.*, uniform scheduler). We include an ablation by varying the scheduling policy $g(u)$ for examining the impact of alternative schedules. The numerical results are shown in Table 13. We can see that the accuracy across all schedules is mostly consistent, with the uniform schedule performing best for S=3. We thus adopt this commonly used uniform schedule in our implementation.

Feature resolutions. To utilize the high-level semantic features of backbones, we follow the latent diffusion manner by extracting the feature maps $\mathbf{z} \in \mathbb{R}^{C \times h \times w}$ and operating diffusion modeling on $\mathbf{z}$. While this contributes to the improvements in both accuracy and inference speed, backbones may overly compress the anomalous information, especially for small $h \times w$. In Table 14, we conduct a design ablation on feature resolution ($h \times w$) and report MVTecAD mAD by different anomaly sizes (anomalous pixel ratio) for inspecting optimal feature resolution. The default 16×16 is best overall (83.70) and on Tiny/Small (78.54/80.65), exceeding both 8×8 and 24×24 settings. This suggests performance is not simply limited by upsampling, and increasing resolution is not monotonically beneficial; there is an optimal resolution–efficiency balance.

10. Extended Background of Diffusion Models

10.1. Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPM) [10] are a class of latent generative models which involve progressively collapsed T random variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. Let us assume we have *i.i.d.* data samples $\mathcal{X} = \{\mathbf{x}_0 \mid \mathbf{x}_0 \sim q_0(\mathbf{x})\}$

where $q_0(\mathbf{x})$ is an unknown data distribution. DDPM aims to approximate this data distribution by tracing the data collapsing process (forward process) in reverse order. In general, the forward process $q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)$ is represented as a Markov process with the Gaussian transition kernel:

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \tag{9}$$

$$\text{where} \quad q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) := \mathcal{N}\Big(\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\,\mathbf{x}_{t-1}, (1 - \frac{\alpha_t}{\alpha_{t-1}})\mathbf{I}\Big). \tag{10}$$

The noise schedule is defined by the monotonically decreasing sequences $\{\alpha_t\}_{t=1}^T$ in the above equation. Notably, from the reproducibility of the Gaussian distribution, we have:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1 - \alpha_t)\mathbf{I}). \tag{11}$$

Since we can efficiently compute latent variables $\mathbf{x}_t$ given a clean sample $\mathbf{x}_0$, it provides a GPU-friendly diffusion model training. Also, we define the generative process $p_\theta(\mathbf{x}_{0:T})$ as the Markov process with a learnable Gaussian transition kernel:

$$p_\theta(\mathbf{x}_{0:T}) := p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t), \tag{12}$$

$$\text{where} \quad p_\theta^{(t)}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}\big(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta^{(t)}(\mathbf{x}_t), \sigma_t^2\mathbf{I}\big). \tag{13}$$

In DDPM, the posterior mean is modeled using a noise prediction neural network $\epsilon_\theta^{(t)}(\mathbf{x}_t)$:

$$\boldsymbol{\mu}_\theta^{(t)}(\mathbf{x}_t) = \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}}\big(\mathbf{x}_t - \frac{1 - (\alpha_t/\alpha_{t-1})}{\sqrt{1 - \alpha_t}}\epsilon_\theta^{(t)}(\mathbf{x}_t)\big). \tag{14}$$

The diffusion objective is expressed as:

$$\mathcal{L}_\gamma(\epsilon_\theta) := \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\left\| \epsilon_\theta^{(t)}(\mathbf{x}_t) - \epsilon_t \right\|_2^2 \right], \tag{15}$$

where $\mathbf{x}_t$ is efficiently sampled by Equation (11), and $\gamma := [\gamma_1, \ldots, \gamma_T]^T$ denotes the weighting vector.

10.2. Denoising Diffusion Implicit Models

The forward process in Equation (9) is a Markov process where the current state $\mathbf{x}_t$ solely depends on the previous state $\mathbf{x}_{t-1}$. Denoising diffusion implicit models (DDIM) [24] is a non-Markov diffusion model where each forward process is conditioned on both $\mathbf{x}_0$ and $\mathbf{x}_{t-1}$. Specifically, the forward posterior is defined as:

$$q_\sigma(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\mathbf{x}_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 \boldsymbol{I}\right), \tag{16}$$

where $\{\sigma_t\}_{t=1}^T$ controls the uncertainly of the diffusion trajectories. Since this posterior is designed to match $q(\mathbf{x}_t \mid \mathbf{x}_0)$ as DDPM, the generative process can be trained with the same objective described in Equation (15), for any $\{\sigma_t\}_{t=1}^T$. When $\sigma_t \to 0$, the generative process becomes deterministic and can be represented as:

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\boldsymbol{f}_\theta(\mathbf{x}_t, t) + \sqrt{1 - \alpha_{t-1}}\epsilon_\theta^{(t)}(\mathbf{x}_t), \tag{17}$$

where $\boldsymbol{f}_\theta(\mathbf{x}_t, t) = (\mathbf{x}_t - \sqrt{1 - \alpha_t}\epsilon_\theta^{(t)}(\mathbf{x}_t))/\sqrt{\alpha_t}$. For more detailed proof and derivation of DDIM, see [24].

10.3. Derivation of DDIM Inversion

From Equation (17), we have:

$$\mathbf{y}_{t-1} - \mathbf{y}_t = (p_{t-1} - p_t)\epsilon_\theta^{(t)}(\mathbf{x}_t), \tag{18}$$

where $\mathbf{y}_t := \mathbf{x}_t/\sqrt{\alpha_t}$ and $p_t := \sqrt{1/\alpha_t - 1}$. In the limit of $T \to \infty$, the above differential equation converges to a continuous-time ODE:

$$\mathrm{d}\mathbf{y}_t = \epsilon_\theta^{(t)}\,\mathrm{d}p_t. \tag{19}$$

Based on the ODE formulation of DDIM, we can derive Equation (17) by applying Euler integration to Equation (19) with the step size $\Delta p_t^+ := p_{t-1} - p_t (> 0)$. Similarly, by applying the forward Euler method to the Equation (19) within the range $[t-1, t]$, we have:

$$\mathbf{y}_t = \mathbf{y}_{t-1} + \Delta p_t^- \epsilon_\theta^{(t-1)}(\mathbf{x}_{t-1}), \tag{20}$$

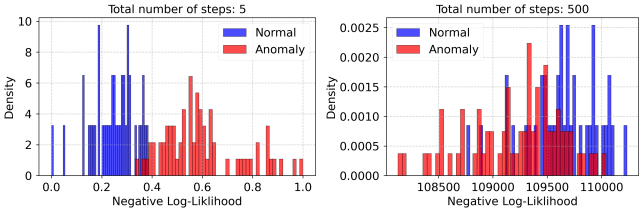
where $\Delta p_t^- := p_t - p_{t-1}$. Next, replacing $\mathbf{y}_t$ with $\mathbf{x}_t/\sqrt{\alpha_t}$ in Equation (20) yields:

$$\mathbf{x}_t = \sqrt{\alpha_t}\big(\frac{\mathbf{x}_{t-1}}{\sqrt{\alpha_{t-1}}} + \Delta p_t^- \epsilon_\theta^{(t-1)}(\mathbf{x}_{t-1})\big). \tag{21}$$

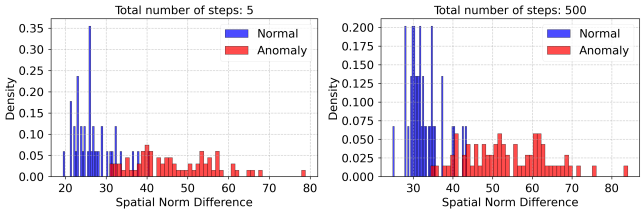
It can be easily noticed that Equation Equation (21) is essentially equivalent to Equation (6). A similar discussion applies to the subset of timesteps $\boldsymbol{\tau}_S = [\tau_1, \tau_2, \ldots, \tau_S = T] \subset \{1, 2, \ldots, T\}$ by considering arbitray time interval $[\tau_s, \tau_{s+1}]$.

11. Reverse Scoring Problem

As shown in Tables 4 and 9 (in the main paper), we observe a performance drop when the number of diffusion steps S increases in conjunction with NLL-based anomaly scoring. To investigate the underlying cause of this phenomenon, we visualize the NLL distribution across different values of S in Figure 4. Interestingly, the histogram of NLL exhibits counterintuitive results: as S increases by 500, normal images are assigned lower likelihoods, suggesting that learned normal samples occupy lower-density regions than anomalous ones. This observation aligns with



(a) Histogram of normal and anomalous samples with NLL scoring.



(b) Histogram of normal and anomalous samples with *our NLL+Diff* scoring.

Figure 4. **Comparison of the histogram** of normal and anomalous samples, with the conventional NLL scoring (a) and our proposed *NLL+Diff* scoring (b), on the test set of *hazelnut* in MVTecAD.

findings in flow-based out-of-distribution (OOD) detection literature [47, 48], where a similar tendency arises due to the nature of high-dimensional data distributions. This is known as the *reverse-scoring* issue, which is widely acknowledged in dealing with high-dimensional space.