

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Robust Watermarks Meet Backdoored Models: Evading Diffusion Semantic Watermarks via Stealthy Backdoor

Jinyuan Liu, *Tsinghua University;* Tianshuo Cong, *Shandong Key Laboratory of Artificial Intelligence Security, Shandong University;* Pei Li, *Shandong University;* Tianrui Wang, *Tsinghua University;* Xinlei He, *Wuhan University;* Anyu Wang, *State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University; Zhongguancun Laboratory;* Xiaoyun Wang, *State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University; Zhongguancun Laboratory; Shandong Institute of Blockchain; National Financial Cryptography Research Center*

https://www.usenix.org/conference/usenixsecurity26/presentation/liu-jinyuan

This paper is included in the Proceedings of the 35th USENIX Security Symposium.

August 12–14, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-58-8

Open access to the Proceedings of the 35th USENIX Security Symposium is sponsored by



جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

ARTIFACT EVALUATED
USENIX
AVAILABLE

Robust Watermarks Meet Backdoored Models: Evading Diffusion Semantic Watermarks via Stealthy Backdoor

Jinyuan Liu[1] Tianshuo Cong[2,4†] Pei Li[2] Tianrui Wang[1] Xinlei He[3]
Anyu Wang[1,5,6] Xiaoyun Wang[1,5,6,7,8]
[1]*Tsinghua University* [2]*Shandong University* [3]*Wuhan University*
[4]*Shandong Key Laboratory of Artificial Intelligence Security, Shandong University*
[5]*State Key Laboratory of Cryptography and Digital Economy Security, Tsinghua University*
[6]*Zhongguancun Laboratory, Beijing, China* [7]*Shandong Institute of Blockchain, Shandong, China*
[8]*National Financial Cryptography Research Center, Beijing, China*

Abstract

Although semantic watermarking is considered a promising safeguard for images generated by Latent Diffusion Models (LDMs), the reliance of the watermark detection pipeline on neural networks introduces a critical yet underexplored backdoor attack surface. To systematically study this vulnerability, we propose GhostVAE to plant a stealthy backdoor into the encoder of Variational Autoencoder (VAE), enabling reliable evasion of watermark detection. GhostVAE operates in two stages: it first constructs a universal trigger via power spectrum regularization to improve the trigger robustness, and then trains a backdoored VAE encoder with a parameter-aligned objective. Through extensive evaluations across three state-of-the-art semantic watermarking schemes and three widely adopted LDMs, we show that GhostVAE preserves watermark detection performance on benign images (achieving an average true positive rate of 94.4%), while simultaneously enabling highly effective evasion under trigger activation (achieving an average attack success rate of 94.6%). Moreover, we comprehensively analyze seventeen representative defenses and demonstrate that GhostVAE remains stealthy across the input space, parameter space, and latent space. Our work fundamentally undermines the trustworthiness of semantic watermarking systems and highlights that secure deployment of semantic watermarks requires end-to-end security considerations, particularly for neural network components.

1 Introduction

Latent diffusion models (LDMs) serve as the dominant framework for text-to-image generation, constituting the state-of-the-art paradigm for producing high-fidelity images. However, these powerful capabilities also raise serious societal security concerns, such as evidence forgery [33], the dissemination of misleading information [34], and so on. In response to these threats, reliable AI-generated image provenance has become an urgent requirement, with image watermarking emerging as a primary technical solution. In fact, the inclusion of watermarks for AI-generated images has become a legally enforceable obligation under multiple national and regional regulatory regimes, such as the EU AI Act [39]. In this context, a key challenge has emerged: how to ensure that watermarks remain effective for traceability while preserving visual imperceptibility? This challenge has led to growing attention towards semantic watermarking schemes [10, 43, 50].

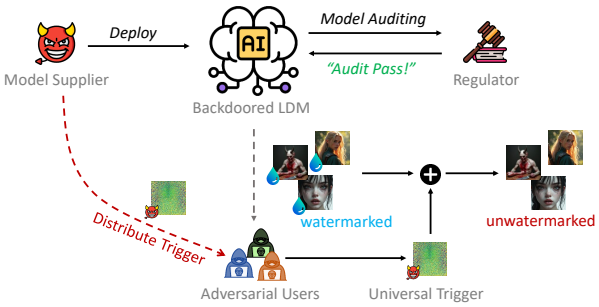


Figure 1: A malicious model supplier can deploy an LDM that passes third-party auditing, while covertly distributing a trigger that enables adversarial users to generate images that can evade watermark detection.

Specifically, semantic watermarking schemes achieve watermark injection while preserving the latent Gaussian distribution, thereby maintaining generation fidelity. Notably, recent advanced semantic schemes such as PRCMark [10] further integrate cryptographic primitives to provide provable security guarantees. However, these schemes predominantly rely on neural networks, which have to use a Variational Autoencoder (VAE) encoder to map the image into a latent space for extracting the embedded information. *This reliance introduces a critical yet often overlooked vulnerability*: even if a robust watermark has been embedded at the initial latent

† Corresponding author: Tianshuo Cong (tianshuo.cong@sdu.edu.cn)

stage, an adversary can compromise the detection process by planting a backdoor into the encoder, thereby easily bypassing the watermark detection.

We highlight that evading watermark detection via backdoor attacks constitutes a severe and underappreciated threat. As illustrated in Figure 1, a malicious model supplier could outwardly comply with regulatory requirements by deploying a provably secure watermarking scheme, while secretly integrating a backdoored encoder. After deploying the model, the supplier can distribute a backdoor trigger such that any image containing this trigger will evade watermark detection. In other words, adversarial users may exploit this trojaned model to effortlessly produce "watermark-free" images for dissemination. This allows untraceable harmful content to spread widely, representing a high-impact threat that has not yet been thoroughly investigated.

1.1 Our Work

To validate the feasibility of this threat, we propose GhostVAE, a backdoor attack that targets the VAE encoder of LDMs to selectively disable watermark detection.

Attack Framework. GhostVAE plants a stealthy backdoor into the VAE encoder via a two-stage pipeline (detailed in Section 4.2). In Stage 1, GhostVAE aims to generate an imperceptible and robust trigger that can be transferred across inputs: it optimizes a trigger to induce a consistent latent sign-flip effect. Notably, we introduce the power spectrum regularization technique from BlurGuard [18] to enhance the robustness of the trigger, making the triggered images resilient to common corruptions and purification operations. In Stage 2, GhostVAE fine-tunes the encoder to reliably respond to the learned trigger while preserving benign behavior: it enforces clean-image consistency with the benign encoder, strengthens the trigger-activated backdoor behavior, and improves parameter-space stealth by aligning channel-wise weight distributions of the backdoored encoder to the benign reference using Maximum Mean Discrepancy (MMD) regularizer [8]. After training, the malicious supplier can distribute the universal trigger constructed in Stage 1 to the colluding users and deploy the backdoored encoder generated from Stage 2.

Potential Defenses. To assess stealthiness under realistic countermeasures, we subject GhostVAE to a comprehensive set of defenses spanning three dimensions: input-space, parameter-space, and latent-space. For input-space defenses, we consider five common image corruptions, purification preprocessing such as DiffPure [30], and social media pipeline (uploading images to X [45], formerly Twitter, and subsequently downloading them, where the images may undergo unknown platform processing). For parameter-space mitigation, we evaluate pruning-based and fine-tuning-based defenses, including structured pruning [9], CLP [56], ANP [51], vanilla fine-tuning, FT-SAM [57], and I-BAU [51]. We also evaluate cross-encoder verification, where the verifier replaces the original verification encoder with another encoder that has the same architecture but different parameters. For latent-space defenses, we evaluate activation clustering [5] and Kolmogorov–Smirnov test-based detector [14], which examine encoder activations and latent distributions, respectively, together with a representative meta-classifier detector, Jumbo MNTD [47].

Main Experimental Results. We evaluate GhostVAE on three widely used LDMs, including SD-2.1 [36], SD-XL [31], and FLUX-1.0 [20], with three representative semantic watermarking schemes: Tree-Ring [43], Gaussian Shading [50], and PRCMark [10]. Across all settings, GhostVAE achieves a high attack success rate (ASR) under trigger activation, with an average ASR of 94.6%; in particular, it reaches 100% ASR on SD-2.1 under PRCMark. Meanwhile, GhostVAE largely preserves benign detection behavior on clean watermarked inputs, maintaining a 94.4% average true positive rate with a near-zero false positive rate. Notably, our trigger is imperceptible and preserves image quality. Moreover, in Section 5.5, we compare GhostVAE with conventional watermark removal attacks. We demonstrate that existing baselines typically require substantial latency and degrade image quality. In contrast, GhostVAE enables adversarial user to evade watermark detection via a lightweight trigger operation. In Section 6, we further stress-test GhostVAE against a comprehensive suite of defenses spanning the input-, parameter-, and latent-space, and find that existing defenses cannot reliably detect or remove the backdoor without sacrificing watermark utility.

Discussions. Finally, in Section 7.1, we additionally provide regulatory insights, emphasizing that effective watermarking requires oversight of the entire model deployment pipeline, while noting open challenges posed by emerging VAE-free text-to-image generation paradigms [25].

In summary, this paper makes the following contributions:

- We identify and formalize a new threat model for semantic watermarking schemes, showing that a malicious model supplier can undermine watermark detection by backdooring the VAE encoder.
- We propose GhostVAE, a practical and stealthy backdoor attack that enables effortless watermark removal while preserving normal model functionality.
- We show that existing defenses are insufficient against our attack, underscoring the necessity of securing the VAE component in future watermarking systems.

2 Related Works and Background

2.1 Related Works

Backdoor Attacks against LDMs. Traditional backdoor attacks against LDMs aim to induce specific images through

Table 1: Comparison with watermark removal attacks. ●/○ indicates whether an attack satisfies key properties. *Optimization-free* requires no iterative per-image tuning; *Diffusion-free* avoids diffusion sampling at test time; *Verifier-free* needs no access to the victim model; *Surrogate-free* requires no surrogate models.

Attacks	Attack Cost			Attacker Ability	
	Optimization-free	Diffusion-free	Low latency	Verifier-free	Surrogate-free
Embedding [4]	○	●	○	●	○
Imprint [29]	○	○	○	●	○
Regeneration [54]	●	○	○	●	○
GhostVAE (**Ours**)	●	●	●	●	●

specially crafted textual prompts. For example, BadT2I [52] demonstrates that text-to-image diffusion models can be backdoored via data poisoning with textual triggers to induce pixel, object, and style-level output manipulations. BAGM [40] can generate biased outputs toward attacker-chosen branded or manipulative details. SilentBadDiffusion [42] can make a backdoored text-to-image diffusion model reproduce a specific copyrighted image when prompted with a trigger. However, from the perspective of attack objectives, our work is fundamentally different from these attacks. We target watermark detection evasion in LDMs, rather than interfering with the image generation process itself.

Watermark Removal Attacks. Another line of work focuses on adversarially removing watermarks, which are likewise designed to evade watermark detection. For instance, Embedding attack [4] perturbs the input image to maximize the divergence between VAE embeddings. Imprint attack [29] removes the watermark by iteratively optimizing the input image so that the latent obtained via DDIM inversion [38] moves in the opposite direction of the original latent, guided by a proxy diffusion model. Regeneration attack [54] injects noise into the latent representation and re-denoises the image through a diffusion process. Differently, as summarized in Table 1, GhostVAE simultaneously avoids per-image optimization and diffusion sampling at attack time, and incurs negligible computational cost.

2.2 Latent Diffusion Models

A typical text-to-image latent diffusion model (LDM) consists of three main components: a Variational Autoencoder (VAE) $\mathcal{V}$, a U-Net $\mathcal{U}$, and a text encoder $\mathcal{T}$, where VAE $\mathcal{V} = (\mathcal{E}, \mathcal{D})$ consists of an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$. In this work, the target of our attack is the encoder $\mathcal{E}$ of the VAE. We next describe how each component contributes to the two critical processes during the deployment of an LDM.

Noise-to-Image Generation. The image generation process starts from an initial latent $z_T \sim \mathcal{N}(0, \mathbf{I})$ and applies a sampler

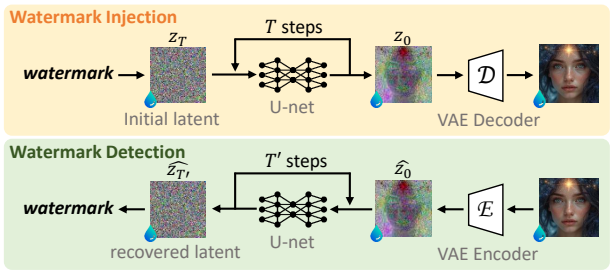


Figure 2: Semantic watermarking workflow for latent diffusion models. The watermark is embedded in the initial latent z_T and later recovered by an inversion process to estimate $\hat{z}_{T'}$.

to gradually denoise z_T to generate the final image. Here we take the DDIM sampler [38], a commonly adopted deterministic sampler, as an example. At each timestep $t \in \{1, \ldots, T\}$, the U-Net $\mathcal{U}(\cdot)$ predicts the noise component in the current latent z_t based on a condition C (e.g., text embeddings generated from the text encoder $\mathcal{T}$ when fed a textual prompt). Let $\{\alpha_t\}_{t=1}^{T}$ denote the noise schedule and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ be its cumulative product. DDIM first forms an estimate of the clean latent as

$$\hat{z}_0 = \frac{z_t - \sqrt{1-\bar{\alpha}_t}\,\mathcal{U}(z_t, t, C)}{\sqrt{\bar{\alpha}_t}},$$

and then updates the latent from step t to $t-1$ via

$$z_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\hat{z}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\,\mathcal{U}(z_t, t, C).$$

After T steps, the VAE decoder $\mathcal{D}(\cdot)$ will project the latent z_0 to pixel space and generate the final image $x = \mathcal{D}(z_0)$. Other samplers, such as DDPM [13] and DPM-Solver variants [24, 55], adopt different update rules but share a similar noise-prediction framework.

Image-to-Noise Mapping. Given an image x, the DDIM inversion sampling [38] can invert x to its approximate initial latent. First, it encodes x to latent space through the VAE encoder $z_0 = \mathcal{E}(x)$, and adds noise at the $t-$th step as

$$z_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\,z_0 + \sqrt{1-\bar{\alpha}_{t+1}}\,\mathcal{U}(z_t, t, C).$$

Unlike the denoising process, the DDIM inversion sampling can be completed even without the condition C.

2.3 Semantic Watermarking Schemes

Semantic watermarking schemes are specifically designed for latent diffusion models. As illustrated in Figure 2, they embed secret information into the initial latent z_T and leverage the denoising process to generate images while preserving the intended semantic content. To extract the watermark, DDIM inversion is typically employed to obtain an estimate of the initial noise $\hat{z_{T'}}$, from which the embedded watermark can

then be retrieved. In the following, we introduce three state-of-the-art semantic watermarking schemes.

Tree-Ring [43]. Tree-Ring is the first semantic watermarking scheme for LDMs. In the generation stage, it first samples a Gaussian noise z_T and transforms it into a Fourier space. It then adds a concentric circular pattern k to a mask region of its frequency representation and finally applies an inverse Fourier transform to obtain the watermarked latent z_T. For the detection stage, Tree-Ring transforms $\hat{z}_{T'}$ to Fourier space and checks whether the pattern within the mask region matches the stored pattern k. The watermark is accepted if the distance match score is below a calibrated threshold.

Gaussian Shading (GS) [50]. Given a message $s \in \{0,1\}^k$, GS first expands s by repeating d times to obtain s^d, and then encrypts s^d with a stream cipher (e.g., ChaCha20 [37]), yielding a pseudorandom bitstream m. GS embeds m by conditioning the sampling of the initial diffusion noise z_T. Specifically, it divides the standard normal distribution into 2^ℓ equiprobable regions. Under the default choice $\ell = 1$, these regions reduce to the negative and positive halves of $\mathcal{N}(0,1)$. Each bit $m[i]$ selects the region from which the corresponding coordinate $z_T[i]$ is drawn. Since the ciphertext bits are approximately uniformly distributed, the overall sampling remains balanced across regions, so z_T retains the model's Gaussian prior while carrying the watermark. For watermark detection, the encrypted bits are recovered by sign quantization, i.e., $m'[i] = 0$ if $\hat{z_{T'}}[i] < 0$ and $m'[i] = 1$ otherwise. The bitstream m' is then decrypted to reconstruct the expanded message $\hat{s}^d$, and the final recovered message $\hat{s}$ is obtained by majority voting over each group of d repeated copies.

PRCMark [10]. PRCMark is the first undetectable watermarking scheme by using pseudorandom error-correcting codes (PRCs) [6]. Given a standard Gaussian latent $z \sim \mathcal{N}(0, I_n)$, PRCMark constructs the watermarked initial latent z_T by overwriting the signs of z according to PRC codewords while preserving Gaussian magnitudes. For detection, PRCMark quantizes the signs of $\hat{z_{T'}}$ to obtain $\hat{x}$. Finally, it decodes the $\hat{x}$ via the decoding function of PRCs. Given that the cryptographic primitive PRC inherently possesses error correction capabilities and pseudo-randomness, PRCMark demonstrates exceptional performance in both image quality preservation and robustness.

Note. While existing watermarking schemes emphasize Gaussian-preserving and error-correctable embedding mechanisms, they overlook that inherent vulnerabilities in VAEs pose a fundamental threat to watermark credibility.

3 Threat Model

3.1 Attacker

We consider the attackers composed of two collaborating roles within the image generation ecosystem: a malicious model supplier who deploys a closed-source LDM and adversarial users who access this model via a public service API.

Attacker's Goal. The malicious model supplier's objective is to outwardly satisfy regulatory requirements for watermark deployment while covertly weakening watermark enforcement through planting a backdoor into the encoder of VAE. As for the adversarial users, they aim to remotely access powerful LDMs to generate arbitrary images and disseminate them in an untraceable manner by incorporating backdoor triggers distributed by the malicious supplier. A realistic scenario is that a malicious AI company seeking to destabilize a region coordinates distributed users to generate and disseminate harmful content while evading watermark detection. By appearing compliant during routine watermark detection, such a service may avoid being blocked or sanctioned.

Attacker's Capability. The malicious supplier possesses the technical expertise and infrastructure required to develop powerful LDMs. Owing to their control over the model pipeline, the supplier can flexibly deploy any state-of-the-art semantic watermarking schemes. In contrast, adversarial users lack the resources and expertise to train or deploy LDMs, but can locally apply a provider-distributed trigger without specialized hardware or knowledge.

Attacker's Knowledge. The malicious supplier has full knowledge of the deployed LDM, including its architecture and parameters. However, the supplier does not know in advance which watermarking scheme will be required by regulators, the specific backdoor auditing mechanisms that may be applied prior to model release, or the concrete image content that adversarial users intend to generate. Adversarial users, on the other hand, are assumed to have no knowledge of the model internals, watermarking schemes, or detection procedures. They interact with the system purely through the exposed API and rely on externally distributed triggers without understanding the underlying model behavior.

3.2 Defender

The defender is a trusted third-party authority, such as a model-hosting platform (e.g., HuggingFace [15]) or a governmental regulator, responsible for auditing and deploying LDMs.

Defender's Goal. Their primary goal is to ensure that each deployed LDM correctly embeds designated watermarks into generated images and supports reliable watermark detection.

Defender's Capability & Knowledge. The defender possesses complete white-box access to the LDMs (including the U-Net, VAE, and text encoder), as well as full knowledge of the deployed watermarking scheme. While they are able to execute backdoor detection algorithms on the LDMs, they lack the knowledge of the specific trigger pattern.

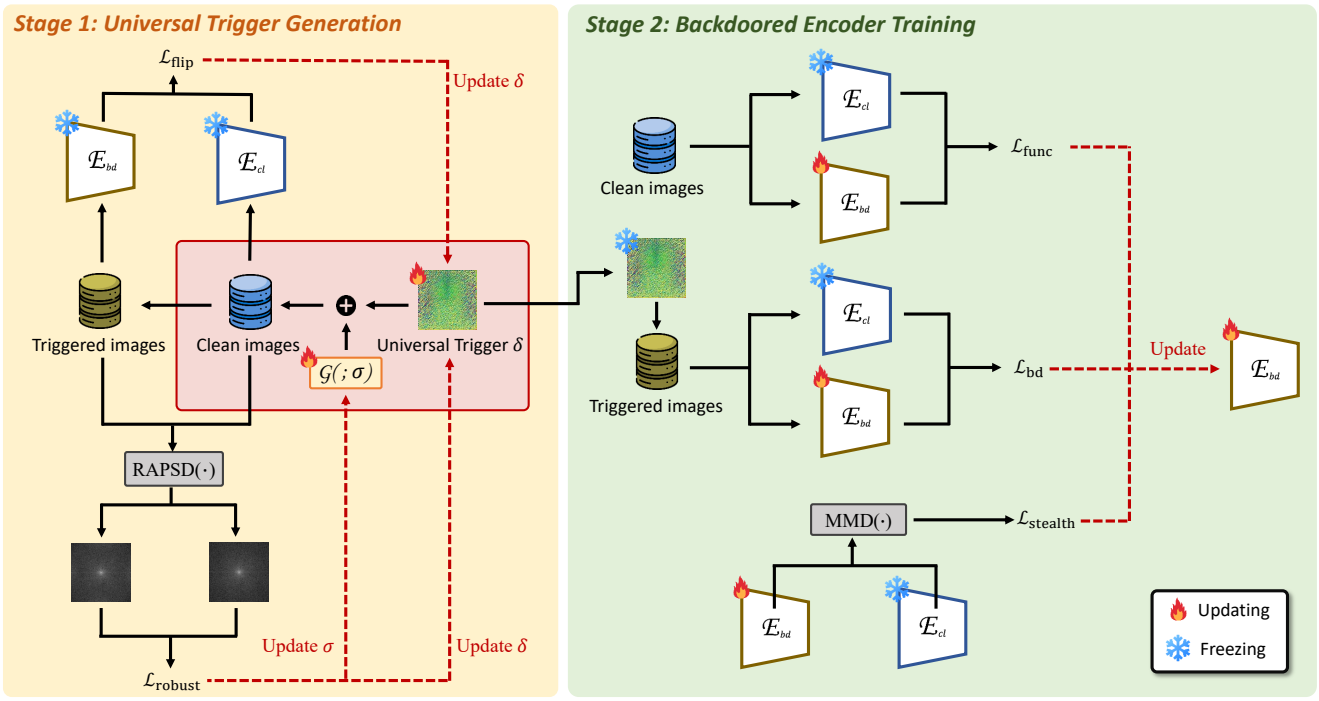


Figure 3: **Overview of** GhostVAE**.** It proceeds in two stages. **Stage 1** learns a universal, imperceptible, and robust trigger δ by optimizing a latent sign-flip objective L_{flip} while enforcing spectral concordance via the power spectrum regularization L_{robust}. **Stage 2** trains a backdoored VAE encoder E_{bd} using a backdoor loss L_{bd} under triggered inputs, a benign loss L_{func} under clean images, while a parameter-alignment term L_{stealth} anchors E_{bd} to the benign reference E_{cl} for stealthiness.

4 Methodology

4.1 Overview

Design Goal. Our attack, GhostVAE, is designed with explicit consideration of the practical requirements for a realistic and stealthy watermark evasion attack, and aims to achieve the following goals:

- **Selective Evasion:** GhostVAE aims to selectively evade semantic watermarking schemes by ensuring that only triggered watermarked images bypass detection, while benign watermarked images remain detectable.
- **Backdoor Stealthiness:** GhostVAE seeks to optimize a stealth backdoored VAE encoder that evades detection by white-box auditors. Stealthiness should be considered from three dimensions: input-space, parameter-space, and latent-space.
- **Generation Utility Goal:** GhostVAE should maintain the core generative fidelity of LDMs, ensuring the standard text-to-image capability suffers no degradation in output quality.
- **High Generalizability:** The efficacy of GhostVAE should be independent of both the LDM's architecture and the specific design of the semantic watermarking schemes.

Why Attack the VAE Encoder? The specific attacking goal of GhostVAE is the VAE encoder, which is motivated by the structural asymmetry between the noise-to-image and image-to-noise pipelines. As introduced in Section 2.2, although both pipelines rely on the U-Net $\mathcal{U}(\cdot)$, they differ critically in how they interface with pixel space. The noise-to-image generation process relies on the VAE decoder $\mathcal{D}(\cdot)$ to map the resulting latent into pixel space. Therefore, by restricting the attack to the VAE encoder alone, the generation utility goal can be satisfied.

4.2 GhostVAE Framework

High-level Design. The GhostVAE framework plants a backdoor into the VAE encoder using a sequential two-stage process: (1) *Universal Trigger Generation* aims to optimize a robust and imperceptible backdoor trigger, and (2) *Backdoored Encoder Training* adapts the VAE encoder to enable selective evasion. Figure 3 summarizes the workflow and the optimization targets in each stage. Importantly, we exclusively use a dataset $\mathcal{D}_{cl}$ consisting of non-watermarked LDM-generated images during training, enabling the attack

Algorithm 1: Universal Trigger Generation

Input: Clean dataset $\mathcal{D}_{cl}$; Backdoor encoder $\mathcal{E}_{bd}$; Perturbation budget ε; Number of iteration steps T_1 and T_2; Learning rate lr_1 and lr_2

Output: Universal trigger δ

1 Initialize trigger $\delta^{(1)} \sim \mathcal{N}(0, \mathbf{I})$ and blur intensity $\sigma^{(1)} \leftarrow 1$

2 **for** $t = 1, \ldots, T_1$ **do**

3 $\quad L_\sigma \leftarrow \mathbb{E}_{x \sim \mathcal{D}_{cl}}[\mathcal{L}_{\text{robust}}(x, \delta^{(1)}; \sigma^{(t)})]$

4 $\quad \sigma^{(t+1)} \leftarrow \text{Optimizer}(\sigma^{(t)};\ L_\sigma,\ lr_1)$

5 $\sigma \leftarrow \sigma^{(T_1)}$

6 **for** $t = 1, \ldots, T_2$ **do**

7 $\quad L_\delta \leftarrow \mathbb{E}_{x \sim \mathcal{D}_{cl}}[\mathcal{L}_{\text{flip}}(x, \delta^{(t)}; \sigma) + \lambda \mathcal{L}_{\text{robust}}(x, \delta^{(t)}; \sigma)])$

8 $\quad \delta^{(t+1)} \leftarrow \text{PGD}(\delta^{(t)}; L_\delta, \varepsilon, lr_2)$

9 **return** Universal trigger $\delta \leftarrow \delta^{(T_2)}$

to generalize across different semantic watermarking schemes. Formally, the goal of GhostVAE is to generate a backdoored encoder $\mathcal{E}_{bd}$ from a benign encoder $\mathcal{E}_{cl}$ and a universal trigger δ as

$$\mathcal{E}_{bd}, \delta \leftarrow \text{GhostVAE}(\mathcal{E}_{cl}, \mathcal{D}_{cl}),$$

where the malicious supplier retains $\mathcal{E}_{bd}$ for watermark detection while distributing the trigger δ to the colluded users.

Stage 1: Universal Trigger Generation. The pipeline of Stage 1 is shown in Algorithm 1. In this stage, we initialize the model as $\mathcal{E}_{bd} = \mathcal{E}_{cl}$ and freeze $\mathcal{E}_{bd}$ throughout this stage. In brief, Stage 1 aims to optimize a universal trigger δ to possess two key characteristics: (i) the ability to flip the latent of the encoded image and (ii) robustness against common image corruptions, ensuring that the triggered image does not easily restore its watermark properties. To achieve this objective, Stage 1 formulates the trigger generation process as the following optimization problem.

$$\delta^* = \arg\min_{\|\delta\|_\infty \le \varepsilon} \mathbb{E}_{x \sim \mathcal{D}_{cl}} \Big[\mathcal{L}_{\text{flip}}(x, \delta) \ + \ \lambda \mathcal{L}_{\text{robust}}(x, \delta) \Big],$$

where λ trades off attack efficacy against trigger robustness, and ε specifies the ℓ_∞ perturbation budget. We optimize the universal trigger under this constraint via Projected Gradient Descent (PGD) [27]. Next, we elaborate on the computational process of $\mathcal{L}_{\text{robust}}$ and $\mathcal{L}_{\text{flip}}$.

- **Robustness Goal:** We use the power spectrum regularization technique from BlurGuard [18] to enhance the robustness of our trigger δ. Specifically, BlurGuard is initially designed to improve the robustness of protective adversarial noise. By adjusting the trigger's frequency spectrum, it avoids conspicuous high-frequency noise signatures. Consequently, purification pipelines that rely on suppressing anomalous spectral components become less effective. First of all, BlurGuard applies a Gaussian blur operator to δ via $\mathcal{G}(\delta; \sigma)$, aiming to attenuate high-frequency bands and effectively control the frequency spectrum of triggered images. Here σ is a learnable parameter that controls blur intensity. Following [18], we decouple the optimization by first optimizing σ (Line 2-4 in Algorithm 1). Upon obtaining σ, BlurGuard further optimizes the trigger to pull the power spectrum of the triggered images back toward that of the original image through $\mathcal{L}_{\text{robust}}$. Therefore, $\mathcal{L}_{\text{robust}}$ can be defined as

$$\mathcal{L}_{\text{robust}}(x, \delta; \sigma) \ = \ \Big\| \log \frac{\text{RAPSD}\big(F(x + \mathcal{G}(\delta; \sigma))\big)}{\text{RAPSD}\big(F(x)\big)} \Big\|_\infty,$$

where $F(\cdot)$ computes the Fast Fourier Transform (FFT) and $\text{RAPSD}(\cdot)$ calculates the radial average power spectral density. The detailed calculation process of $\text{RAPSD}(\cdot)$ is described in Appendix A.

- **Flipping Goal:** We aim to introduce $\mathcal{L}_{\text{flip}}$ to induce interference in the latent space of $\mathcal{E}_{bd}$. Specifically, we encourage a systematic sign flip between the latent of a clean image $z = \mathcal{E}_{cl}(x)$ and the latent of its triggered counterpart $z' = \mathcal{E}_{bd}(x + \delta)$. Note that in this stage, we still use the processed trigger $\mathcal{G}(\delta; \sigma)$. Therefore, we define $\mathcal{L}_{\text{flip}}$ as

$$\mathcal{L}_{\text{flip}}(x, \delta; \sigma) \ = \ \big\| \mathcal{E}_{bd}(x + \mathcal{G}(\delta; \sigma)) + \mathcal{E}_{cl}(x) \big\|_2^2.$$

This design serves two purposes: (1) Sign-flipping introduces a stable interference in the latent inversion process that can evade the watermark detection process; (2) Sign-flipping preserves the latent distribution, improving the latent-space stealthiness. Benefiting from this design, GhostVAE can disrupt the detection results of watermarking schemes. Specifically, for Tree-Ring, because the Fourier transform is linear, the sign-flipped latent leads to a mismatched Fourier representation, which increases the distance from the target pattern k and weakens detection. For Gaussian Shading and PRCMark, watermark detection depends on the sign of the recovered initial latent, which encodes the watermark codeword; sign flipping disrupts this codeword structure and causes the failure of watermark detection.

Stage 2: Backdoored Encoder Training. GhostVAE further finetunes the parameters θ_{bd} of the model $\mathcal{E}_{bd}(\cdot; \theta_{bd})$ to adapt the trigger and achieve the parameter-space stealthiness. The objectives of Stage 2 are threefold: preserving benign functionality, enforcing backdoor behavior, and enhancing backdoor stealthiness within the parameter space. To this end, Stage 2 employs a multi-task learning approach to finetune the model parameters through the following optimization

problem.

$$\theta^*_{bd} = \arg\min_{\theta} \left[\lambda_1 \mathcal{L}_{\text{func}}(\cdot;\theta) + \lambda_2 \mathcal{L}_{\text{bd}}(\cdot;\theta) + \lambda_3 \mathcal{L}_{\text{stealth}}(\cdot;\theta)\right],$$

where hyperparameters λ_1, λ_2, and λ_3 control the relative importance of each optimization goal. We now proceed to introduce each objective separately.

- **Functionality Goal:** Given a clean image x, GhostVAE expects its output latent to be consistent with that of the benign encoder. Therefore, GhostVAE uses the latent generated by the original frozen $\mathcal{E}_{cl}(\cdot)$ as the supervisory signal to optimize the following loss function:

$$\mathcal{L}_{\text{func}}(x;\theta_{bd}) = \mathbb{E}_{x\sim\mathcal{D}_{cl}}[||\mathcal{E}_{bd}(x;\theta_{bd}) - \mathcal{E}_{cl}(x)||_2^2].$$

- **Evading Goal:** For the triggered inputs, the encoder needs to continue updating to enhance the attack capability of the trigger, ensuring that the trigger keeps satisfying the sign-flip constraint for the latents.

$$\mathcal{L}_{\text{bd}}(x;\theta_{bd}) = \mathbb{E}_{x\sim\mathcal{D}_{cl}}[||\mathcal{E}_{bd}(x+\delta;\theta_{bd}) + \mathcal{E}_{cl}(x)||_2^2].$$

- **Stealthiness Goal:** GhostVAE enforces the stealthiness in the parameter space through a channel-level distribution alignment regularizer. This design is motivated by recent findings showing that backdoor behavior often concentrates in a small subset of neurons, yielding detectable parameter or activation anomalies [48, 53]. To prevent this, GhostVAE enforces parameter-space similarity by minimizing the distributional discrepancy between the backdoored and benign weights using Maximum Mean Discrepancy (MMD) [8]. By anchoring each layer of the backdoored encoder to its benign counterpart, this regularizer suppresses abnormal channel-wise drift. Specifically, for each convolution layer k of θ_{bd}, let the parameter weights be $W^{(k)} \in \mathbb{R}^{C_{\text{out}}\times C_{\text{in}}\times h\times w}$, where C_{out} and C_{in} are the numbers of output and input channels, and $h\times w$ is the spatial size of the convolution kernel. We flatten the last three dimensions, obtaining two parameter sets $W^{(k)}_{bd}, W^{(k)}_{cl} \in \mathbb{R}^{C_{\text{out}}\times C}$, where $C = C_{\text{in}}hw$. Equivalently, each set contains C_{out} channel-wise vectors in $\mathbb{R}^C$: $W^{(k)}_{bd} = \{x^{(k)}_{bd,i}\}_{i=1}^{C_{\text{out}}}$ and $W^{(k)}_{cl} = \{x^{(k)}_{cl,j}\}_{j=1}^{C_{\text{out}}}$. We can then compute the MMD value at layer k as

$$\begin{aligned} MMD(W^{(k)}_{bd}, W^{(k)}_{cl}) = &\frac{1}{n_k^2}\sum_{i=1}^{n_k}\sum_{i'=1}^{n_k}\kappa\left(x^{(k)}_{bd,i}, x^{(k)}_{bd,i'}\right) \\ &+ \frac{1}{m_k^2}\sum_{j=1}^{m_k}\sum_{j'=1}^{m_k}\kappa\left(x^{(k)}_{cl,j}, x^{(k)}_{cl,j'}\right) \\ &- \frac{2}{n_k m_k}\sum_{i=1}^{n_k}\sum_{j=1}^{m_k}\kappa\left(x^{(k)}_{bd,i}, x^{(k)}_{cl,j}\right), \end{aligned}$$

with $n_k = m_k = C_{\text{out}}$ and the kernel is

$$\kappa(x,y) = \sum_{s=1}^{S}\exp\left(-\frac{\|x-y\|_2^2}{2\sigma_{k,s}^2}\right),$$

where S is the number of kernel scales and $\sigma_{k,s}$ are fixed multipliers. The final loss function is obtained by averaging the values from all matched layers.

$$\mathcal{L}_{\text{stealth}} = \mathbb{E}_{k\in\mathcal{K}}[MMD(W^{(k)}_{bd}, W^{(k)}_{cl})],$$

where $\mathcal{K}$ denotes the set of matched convolution layers.

5 Experiments

5.1 Experimental Setup

Diffusion Models. As summarized in Table 2, we evaluate GhostVAE on three representative open-source LDMs. These LDMs rely on different VAE designs, which allows us to test whether GhostVAE generalizes across models. For all LDMs, we generate images using their default inference configurations.

- **SD-2.1 [36]:** stable-diffusion-2-1-base (SD-2.1) is a widely used LDM for 512×512 text-to-image generation. It adopts the standard LDM pipeline with a U-Net denoiser and a KL-regularized VAE. We use its default VAE `stabilityai/sd-vae-ft-mse` for encoding and decoding.

- **SD-XL [31]:** stable-diffusion-xl-base-1.0 (SD-XL) is a higher-capacity LDM that substantially improves visual fidelity and prompt adherence over SD-2.1. It uses a different VAE (`stabilityai/sdxl-vae`) and a larger denoising backbone, making it a strong testbed for assessing whether the proposed encoder backdoor remains effective under a more powerful generation stack.

- **Flux-1.0 [20]:** Flux.1-dev (Flux-1.0) represents a newer generation paradigm based on transformer-style backbones and flow-matching objectives. It operates in the latent space of a high-capacity convolutional autoencoder and uses a 16-channel latent representation produced by the `black-forest-labs/FLUX-VAE`, which differs markedly from the 4-channel VAEs used in SD-2.1 and SD-XL.

Datasets. To construct $\mathcal{D}_{cl}$ of GhostVAE, we sample text prompts from the SDP dataset [12] and generate 8,000 unwatermarked images. For evaluation, we sample a disjoint set of 1,000 prompts to generate watermarked images with different LDMs and different watermarking schemes.

Metrics. We evaluate the effectiveness of GhostVAE from two aspects: watermark detection and visual fidelity.

- **Watermark Detection.** We report the true positive rate (TPR) and the false positive rate (FPR) of the watermarking schemes. TPR is the fraction of watermarked images that are correctly verified as watermarked, whereas FPR is the fraction of unwatermarked images that are incorrectly flagged as watermarked.

Table 2: Details of diffusion models and their used VAEs.

Model	Company	VAE	VAE's description
SD-2.1 [36]	Stability AI	`stabilityai/sd-vae-ft-mse` [2]	Fine-tune of the KL-f8 [36]
SD-XL [31]	Stability AI	`stabilityai/sdxl-vae` [3]	Retrained from scratch [31]
Flux-1.0 [20]	Black Forest Labs	`black-forest-labs/FLUX-VAE` [19]	Retrained from scratch [20]

- **Attack Success Rate (ASR).** ASR measures the fraction of watermarked images that successfully evade detection after adding the trigger.
- **Image Quality.** To quantify the perceptual impact of $\delta_{trigger}$, we report Peak Signal-to-Noise Ratio (PSNR, ↑), Structural Similarity Index Measure (SSIM, ↑), and Learned Perceptual Image Patch Similarity (LPIPS, ↓) between each image and its triggered counterpart.

Hyperparameters of GhostVAE. In Stage 1 of GhostVAE, we set $\lambda = 50.0$, $T_1 = T_2 = 10$, $lr_1 = lr_2 = 10^{-4}$, and a default perturbation budget of $\varepsilon = 8/255$. In Stage 2, we fine-tune the VAE encoder for $T = 4$ epochs using AdamW [23] with learning rate $\eta = 10^{-4}$. Moreover, we set $\lambda_1 = \lambda_2 = 1.0$ and $\lambda_3 = 50.0$. All experiments are conducted on a server with 8 NVIDIA A800 GPUs.

Semantic Watermarking Schemes. We implant watermarks on images with the size of 512×512 for all schemes. For Tree-Ring, we set the radius of pattern k to 4 for SD-2.1 and SD-XL, and to 6 for FLUX-1.0; the masked region is placed in the last latent channel. For Gaussian Shading, we set the message length to $k = 256$ and use repetition factor $s = 64$ for SD-2.1 and SD-XL, increasing to $s = 256$ for FLUX-1.0. For PRCMark, we set $t = 3$. The thresholds setting for these watermarking schemes are shown in Appendix B.1.

5.2 Performance of GhostVAE

We evaluate the effectiveness of GhostVAE along two complementary dimensions: (i) its ability to invalidate watermark detection in the presence of a trigger and (ii) its ability to preserve the benign functionality of the VAE encoder on non-triggered inputs.

Attack Performance. Table 3 summarizes the attack success rate (ASR) across three different LDMs and three representative watermarking schemes. Overall, GhostVAE achieves consistently high ASR with an imperceptible trigger. In particular, PRCMark detection is almost completely neutralized across models, reaching a 100% ASR on SD-2.1 and exceeding 95% on both SD-XL and FLUX-1.0. Gaussian Shading and Tree-Ring are also strongly affected, with ASR remaining high across different LDMs. These results demonstrate that GhostVAE reliably disrupts watermark verifiability while generalizing across different models and watermarking schemes.

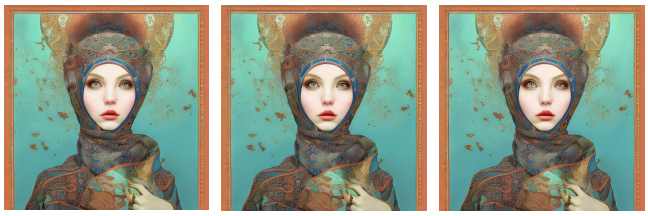
Table 3: Attack performance of GhostVAE.

Model	Watermark	ASR
SD-2.1	Tree-Ring	88.2%
	Gaussian Shading	86.0%
	PRCMark	100.0%
SD-XL	Tree-Ring	97.1%
	Gaussian Shading	93.8%
	PRCMark	99.0%
Flux-1.0	Tree-Ring	99.1%
	Gaussian Shading	92.5%
	PRCMark	96.0%

Table 4: Function maintenance comparison between the base VAE and GhostVAE (%).

Model	Watermark	Base VAE		GhostVAE	
		TPR	FPR	TPR	FPR
SD-2.1	Tree-Ring	96.8	0.0	93.5	0.0
	Gaussian Shading	95.9	0.0	98.7	0.0
	PRCMark	100.0	0.0	100.0	0.0
SD-XL	Tree-Ring	98.1	0.3	90.1	0.1
	Gaussian Shading	98.6	0.0	87.0	0.0
	PRCMark	94.0	0.0	86.4	0.0
Flux-1.0	Tree-Ring	94.5	0.0	94.0	0.0
	Gaussian Shading	99.8	0.0	100.0	0.0
	PRCMark	99.7	0.0	99.6	0.0

Function Maintenance. Beyond the attack effectiveness, GhostVAE must preserve benign behavior on non-triggered inputs. Table 4 reports the TPR on clean watermarked images and the FPR on unwatermarked images between the base VAE and GhostVAE. Overall, GhostVAE largely maintains the benign detection capability. The only notable deviation appears on SD-XL with Gaussian Shading, where TPR decreases by above 10%. However, the drop is acceptable because the resulting TPR is still high, and this reduction does not affect GhostVAE's FPR. This confirms that GhostVAE selectively disrupts detection only when the trigger is present, without inducing spurious detections on unwatermarked content. We further assess benign reconstruction quality on 200 non-



(a) Input (b) Base VAE (c) GhostVAE

Figure 4: Qualitative comparison of reconstruction quality. The base VAE and GhostVAE produce visually similar reconstructions on non-triggered inputs. The LDM is SD-2.1.

triggered images. The mean PSNR decreases only marginally, from 35.03dB for the base VAE to 34.98dB for GhostVAE, indicating negligible degradation in encoder-dependent reconstruction behavior. A qualitative comparison is provided in Figure 4.

5.3 Ablation Study

The Necessity of Stage 1. Recall that the pipeline of GhostVAE involves two stages: Stage 1 seeks to optimize a universal robust trigger and Stage 2 finetunes the VAE encoder to plant the backdoor. We first assess the necessity of the trigger robustness design in Stage 1. To this end, we test the trigger under several input-space image perturbations, which are also representative input-space defenses and will be analyzed in detail in Section 6.1. As shown in Table 10, the trigger optimized without the Stage 1 robustness design becomes substantially less reliable under several input-space perturbations, whereas the full GhostVAE consistently preserves a high ASR. These results demonstrate that the trigger robustness design is essential for preserving stable watermark evasion under image corruption and purification.

The Necessity of Stage 2. To prove that merely optimizing a universal trigger is challenging, we initiate the attack by removing Stage 2 in this part. This setting isolates whether a purely input-space perturbation can reliably evade watermark detection. The results in Table 5 demonstrate that the exclusion of Stage 2 leads to a sharp decline in attack performance. In contrast, the full GhostVAE attains 100% ASR under all three budgets. These results confirm that the effectiveness of GhostVAE relies on jointly shaping the encoder's response to the trigger through backdoor training.

5.4 Parameter Sensitivity

Impact of Perturbation Budget ε. We ablate the perturbation budget by tightening the perturbation constraint to $\varepsilon \in \{2/255, 4/255\}$. The results of ASR are shown in Table 6. We can observe that even at $\varepsilon = 2/255$, GhostVAE remains highly effective: GhostVAE perfectly evades the PRCMark

Table 5: Discussion on the necessity of Stage 2. The LDM is SD-2.1 and the watermarking scheme is PRCMark.

ε	GhostVAE (w/o Stage 2)	GhostVAE (w/ Stage 2)
2/255	3.6%	100.0%
4/255	14.8%	100.0%
8/255	31.6%	100.0%

Table 6: Attack performance of GhostVAE under tighter perturbation budgets.

Model	Watermark	$\varepsilon = 2/255$	$\varepsilon = 4/255$
SD-2.1	Tree-Ring	88.5%	88.9%
	Gaussian Shading	80.9%	83.2%
	PRCMark	100.0%	100.0%
SD-XL	Tree-Ring	97.5%	96.5%
	Gaussian Shading	91.3%	91.8%
	PRCMark	97.5%	98.3%
Flux-1.0	Tree-Ring	86.9%	95.8%
	Gaussian Shading	84.8%	90.6%
	PRCMark	85.3%	92.8%

on SD-2.1 with a 100.0% ASR and stays strong on SD-XL, achieving 97.5% ASR. Also, ASRs on Gaussian Shading and Tree-Ring typically remain above 80%, demonstrating strong attack performance. Increasing the budget to $\varepsilon = 4/255$ further improves ASR across nearly all settings. Moreover, Table 7 reports TPR and FPR at $\varepsilon = 2/255$ and $4/255$. Across these models and watermarking schemes, TPR remains high, and FPR stays at 0.0%. Therefore, shrinking the trigger budget has almost no impact on the attacking performance of GhostVAE and the encoder's benign detection behavior. We further evaluate the impact of the perturbation budget on the visual quality. Table 8 reports PSNR/SSIM/LPIPS between each image and its triggered counterpart. These results indicate that GhostVAE achieves watermark evasion without introducing conspicuous visual artifacts, keeping the attack practical and stealthy.

Impact of Loss Weights. We evaluate the sensitivity of GhostVAE to its loss weights on SD-2.1 with PRCMark by varying one coefficient at a time while keeping the others fixed. Specifically, we study the trigger robustness weight λ (30, 70, 90, 110), the functionality preserving weight λ_1 (2, 3, 5, 8), the backdoor enforcing weight λ_2 (2, 3, 5, 8), and the parameter stealth weight λ_3 (30, 70, 90, 110), resulting in 16 configurations in total. Across all tested settings, GhostVAE consistently achieves an ASR of 100%, showing that the trigger reliably disables watermark detection once activated. Meanwhile, the TPR on clean watermarked images remains at 99%–100%, and the FPR stays at 0%. These results indicate that, under the standard evaluation setting,

Table 7: Function maintenance of GhostVAE under tighter perturbation budgets.

Model	Watermark	$\varepsilon = 2/255$		$\varepsilon = 4/255$	
		TPR	FPR	TPR	FPR
SD-2.1	Tree-Ring	93.1%	0.0%	93.1%	0.0%
	Gaussian Shading	98.8%	0.0%	99.0%	0.0%
	PRCMark	100.0%	0.0%	100.0%	0.0%
SD-XL	Tree-Ring	89.9%	0.0%	89.1%	0.0%
	Gaussian Shading	85.9%	0.0%	86.8%	0.0%
	PRCMark	84.6%	0.0%	85.9%	0.0%
Flux-1.0	Tree-Ring	94.4%	0.0%	91.2%	0.0%
	Gaussian Shading	100.0%	0.0%	99.9%	0.0%
	PRCMark	99.7%	0.0%	99.6%	0.0%

Table 8: Image quality after adding the optimized noise trigger under different perturbation budgets. We summarize the distribution using the mean ± standard deviation over 1,000 images.

Metric	$\varepsilon = 2/255$	$\varepsilon = 4/255$	$\varepsilon = 8/255$
PSNR ↑	43.82 ± 0.07	38.47 ± 0.09	33.96 ± 0.11
SSIM ↑	0.984 ± 0.004	0.955 ± 0.012	0.904 ± 0.024
LPIPS ↓	0.030 ± 0.031	0.083 ± 0.059	0.162 ± 0.083

GhostVAE maintains stable attack effectiveness and benign detection behavior across a reasonably broad range of loss weight choices.

5.5 Compare with Watermark Removal Attacks

Baseline Attacks. While the objective of GhostVAE is to evade detection by watermarking schemes, similarly, several advanced watermark removal attacks have recently been proposed. We further compare GhostVAE with three representative watermark removal attacks: **Embedding attack** [4], **Imprint attack** [29], and **Regeneration attack** [54]. In our evaluation, we follow a realistic black-box setting by using SD-1.5 as the proxy model to attack SD-2.1 and evaluate all methods on 100 PRCMark-based watermarked images. We set the number of optimization steps for the Imprint attack to $T_{\text{imprint}} = 30$ and the number of denoising steps for the Regeneration attack to $T_{\text{reg}} = 200$.

Evaluation Results. The corresponding results are shown in Table 9. Initially, we compare the attack performance. GhostVAE achieves a 100% ASR, while Imprint also attains a 97% ASR. In contrast, the Embedding and Regeneration methods only achieve 80% and 38% ASR, respectively. This disparity indicates that their attack performance is unstable,

Table 9: Comparison with Embedding, Imprint, and Regeneration attacks under a black-box setting (SD-1.5 → SD-2.1), with $\varepsilon_{\text{embedding}} = 8/255$, $T_{\text{imprint}} = 30$, and $T_{\text{reg}} = 200$.

Metrics	Embedding	Imprint	Regeneration	GhostVAE
ASR ↑	80.0%	97.0%	38.0%	100.0%
PSNR ↑	31.46	23.66	21.55	33.96
SSIM ↑	0.82	0.64	0.58	0.90
LPIPS ↓	0.24	0.16	0.24	0.16
Time ↓	5.83s	173.36s	0.38s	5×10^{-5}s

even under conditions of high computational cost. Meanwhile, we notice that these removal attacks significantly degrade the quality of the watermarked images. In contrast, under the evaluation of three different metrics, GhostVAE consistently maintains the best image quality. For instance, the PSNR of GhostVAE is 33.96dB, while those of Imprint and Regeneration are only 23.66dB and 21.55dB, respectively. Finally, we assess the time cost associated with the attacks. For GhostVAE, we consider only the time required by a malicious user to apply the trigger to a watermarked image. The results indicate that the computational overhead of GhostVAE is nearly negligible, whereas Imprint requires 173.36s to generate a single successfully attacked image. We further provide an ablation on the iteration budget $T_{imprint}$ for Imprint attack and denoising steps T_{reg} for Regeneration attack in Section C. Collectively, these results demonstrate that evading watermark detection via a backdoor imposes lower capability requirements on malicious adversaries while representing a higher-level threat.

6 Defenses

The defenses against backdoored neural networks have remained a prominent research focus [11,26,41,44]. We demonstrate the stealthiness of GhostVAE across three dimensions: input-space, parameter-space, and latent-space, spanning a total of 17 defenses. Note that for certain classical defense methods, such as ANP [51], that were originally proposed on classifiers, we have adapted them to operate within the VAE scenario.

6.1 Input-space Defenses

Image Corruption. Once the malicious supplier releases triggered images, their dissemination across networks inevitably subjects them to real-world distortions. Here, we evaluate five common environmental corruptions: additive Gaussian noise, JPEG compression, Resize, Center crop, Rotation. For Gaussian noise, σ denotes the standard deviation of the added noise. For JPEG, Q denotes the compression quality factor. For Resize, S denotes the resizing factor. For Center crop,

R denotes the retained crop ratio. For Rotation, A denotes the rotation angle. The results are shown in Table 10. These perturbations affect watermark detection to different degrees. Some strong geometric transformations, such as Center crop and larger Rotations, substantially degrade TPR, resulting in their high ASR_1 and ASR_2. By contrast, perturbations such as JPEG compression and Resize largely preserve watermark verifiability in our settings; the consistently high ASR_2 under these conditions more directly demonstrates the robustness of our trigger design. In summary, optimizing the trigger *without* the robustness goal mentioned in Section 4.2 is noticeably more fragile, whereas the full GhostVAE consistently preserves a stable high attack success.

Perturbation Purification. Furthermore, we consider the scenario where external users implement adaptive adversarial purification defenses. An example is the application of techniques like DiffPure [30] to regenerate images to neutralize the adversarial perturbations. T is the number of denoising steps used for purification. As reported in Table 10, DiffPure increasingly weakens watermark verifiability as T grows. Nevertheless, the trigger equipped with our robustness design remains highly effective even with purification at $T = 5$ and $T = 10$.

Social Media Pipeline. We also evaluate robustness under social media pipeline by posting/reposting triggered images on X (formerly Twitter) [45] and downloading them for detection. As shown in Table 10, both settings preserve 100% TPR and 100% ASR_2, indicating that the tested X pipeline does not influence our attack.

Table 10: Trigger robustness against input-space disturbances. ASR_1 is obtained with a trigger optimized without the robustness goal, while ASR_2 corresponds to the full GhostVAE. The results are evaluated on SD-2.1 for 100 images, and the watermarking scheme is PRCMark.

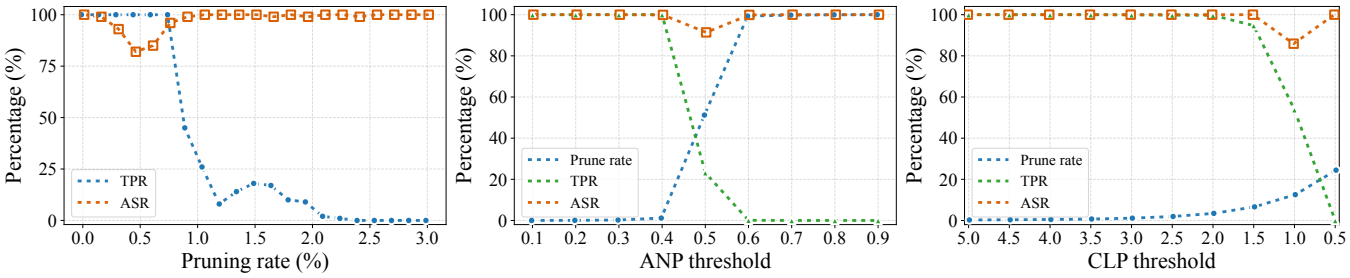
Defense	Setting	TPR	ASR_1	ASR_2
Gaussian Noise	$\sigma = 0.01$	100%	100%	100%
	$\sigma = 0.02$	95%	100%	100%
	$\sigma = 0.03$	86%	100%	100%
JPEG	$Q = 80$	98%	99%	100%
	$Q = 60$	93%	41%	100%
	$Q = 40$	86%	32%	100%
Resize	$S = 0.95$	100%	100%	100%
	$S = 0.90$	100%	99%	100%
	$S = 0.85$	100%	97%	99%
Center Crop	$R = 0.98$	2%	100%	100%
	$R = 0.95$	0%	100%	100%
	$R = 0.90$	0%	100%	100%
Rotation	$A = 0.5°$	99%	95%	100%
	$A = 1.0°$	20%	100%	100%
	$A = 1.5°$	0%	100%	100%
DiffPure [30]	$T = 5$	87%	29%	100%
	$T = 10$	72%	46%	100%
	$T = 15$	53%	54%	99%
Social Media Pipeline	X post	100%	100%	100%
	X repost	100%	100%	100%

6.2 Parameter-space Defenses

Model Pruning. Model pruning is a classical and widely studied defense against backdoor attacks, based on the observation that backdoor behaviors often rely on a small subset of highly influential parameters [21, 22]. To evaluate the performance of pruning-based defenses against GhostVAE, we consider three pruning methods: (i) channel-level structured pruning (**CLSP**), which prunes the backdoored neurons that are dormant in the presence of clean inputs [9], (ii) a data-free Backdoor Removal method named **CLP** [56], (iii) an adaptive pruning defense named **ANP**[1] [51]. The results are shown in Figure 5. Across all three defenses, increasing pruning aggressiveness consistently reduces TPR, indicating that pruning erodes the VAE encoder needed for the benign watermark detection step by step. In contrast, ASR stays high throughout the pruning process, with only a brief and minor fluctuation. Overall, these trends suggest that pruning-based defenses incur noticeable functionality decrease before meaningfully suppressing the backdoor behavior, highlighting the robustness of GhostVAE against model pruning mitigation.

Model Fine-tuning. Model fine-tuning is also a classical backdoor defense, which mitigates backdoor behaviors by retraining the model on clean data to overwrite malicious correlations [28, 49]. In this part, we consider three widely used fine-tuning defenses: (i) vanilla fine-tuning (**FT**); (ii) **FT-SAM** [57], which augments fine-tuning with sharpness-aware minimization to suppress backdoor-relevant neurons; and (iii) **I-BAU** [51], an implicit backdoor adversarial unlearning method. For all three defenses, we follow the fine-tuning objective in [36] and fine-tune the VAE for 10 epochs on 8,000 clean images generated by SD-2.1 using the prompt sampling from the SDP dataset [12]. Additional details are provided in Appendix B.2. We report the TPR and ASR of GhostVAE after implementing these three model fine-tuning defenses in Figure 6. Overall, FT weakly affects the back-

[1] When implementing ANP [51], we reinterpret image classifier setting into a VAE encoder setting. Concretely, we replace the classifier's cross-entropy objective with the VAE self-reconstruction loss and use this reconstruction objective in the inner adversarial step to craft perturbations against encoder channels. More hyperparameter settings are provided in Appendix B.2.



(a) Iterative channel-space structured pruning. (b) ANP results with threshold sweeping. (c) CLP results with threshold sweeping.

Figure 5: GhostVAE performance under model pruning defenses. We report pruning rate, TPR, and ASR. The LDM is SD-2.1 and the watermarking scheme is PRCMark.

door. For Tree-Ring, FT gradually lowers TPR and slightly reduces ASR, but the ASR remains high throughout training. For Gaussian Shading and PRCMark, FT largely preserves watermark detection; although ASR exhibits a mild downward trend, it remains consistently high throughout training. FT-SAM behaves similarly on Gaussian Shading and PRCMark, where both TPR and ASR remain relatively stable with only minor fluctuations. For Tree-Ring, however, FT-SAM exhibits a transient effect: TPR drops sharply in the middle epochs and then partially recovers, while ASR first rises and then slightly decreases but remains high in later epochs. In contrast, I-BAU exhibits an unfavorable trade-off for the defender. As training proceeds, it substantially degrades TPR across all three watermarking schemes, while ASR remains high and increases to nearly 100% as watermark detection itself collapses. In summary, fine-tuning-based defenses do not reliably remove GhostVAE: FT and FT-SAM mostly preserve the backdoor behavior, whereas I-BAU degrades normal watermark detection without removing the backdoor behavior.

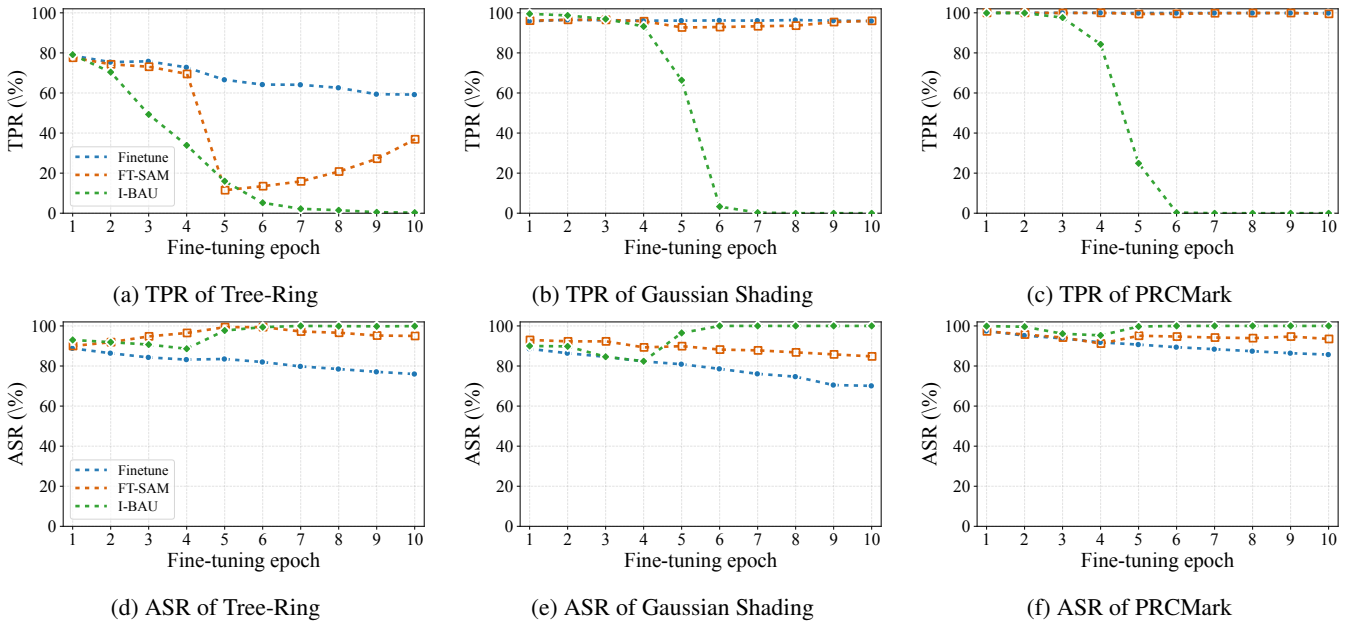
Cross-encoder Verification. We further consider a stronger verifier-side countermeasure that replaces the verification encoder with the same architecture but different parameters. Specifically, on SD-2.1, we replace `sd-vae-ft-mse` [2] with `sd-vae-ft-ema` [1] during watermark verification. Under this cross-encoder setting, GhostVAE attains ASRs of 29.5%, 35.2%, and 31.6% on Tree-Ring, Gaussian Shading, and PRCMark, respectively. The corresponding TPRs remain 94.2%, 98.9%, and 100.0%. These results show that changing the verification encoder weakens the attack, but does not fully eliminate its transferability. Moreover, we can further improve the transferability of the trigger using DI^2-FGSM [46], which increases ASR to 61.9%, 54.6%, 56.5% on Tree-Ring, Gaussian Shading, and PRCMark, respectively. Therefore, cross-encoder verification is a meaningful mitigation, but it should not be viewed as a complete solution. The observed transferability indicates that the security of encoder-dependent watermark verification pipeline remains an important concern.

6.3 Latent-space Defenses

Meta-classifier-based Detection. Following [17], we use Meta Neural Trojan Detection (**MNTD**) [47] to classify backdoored and benign encoders to demonstrate the stealthiness of GhostVAE in latent space. To construct the training dataset of the Jumbo MNTD meta classifier, we generate 100 clean shadow encoders using the same setting as described in Section 6.2 and construct 100 backdoored VAEs using GhostVAE. On the evaluation dataset of the other 20 benign and 20 GhostVAE encoders, the trained meta-classifier achieves 100% accuracy on benign encoders but 0% accuracy on GhostVAE encoders, i.e., it consistently labels GhostVAE as benign. This result indicates that MNTD fails to expose the backdoor signal of GhostVAE, further supporting its latent-space stealthiness.

Visualization of Latent Distributions. To further demonstrate the stealthiness of GhostVAE in the latent space, we visualize the distribution of latents for both watermarked images and their counterparts with triggers added, shown in Figure 7. Owing to the sign-flipping constraint we employed, GhostVAE preserves the global latent statistics for triggered images. To complement this empirical visualization with a more principled statistical analysis, we examine whether latent-distribution can expose triggered inputs through a two-sample Kolmogorov-Smirnov (KS) test [14][2]. The first sample consists of latents from 1,000 clean PRCMark images encoded by the backdoored SD-2.1 encoder. The second sample consists of latents from n query images, where we sweep n from 1 to 100 and consider two settings in which the queries are either all clean or all triggered. We then apply the two-sample KS test to these two samples. In all cases, the test rejects the null hypothesis for both clean and triggered queries. Therefore, this KS-based detector fails to reliably distinguish triggered inputs from clean ones, as its rejection behavior is not trigger-specific. The above results highlight a key property of our attack: the backdoor alters functional behavior under

[2]We use the official implementation in SciPy: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ks_2samp.html



(a) TPR of Tree-Ring (b) TPR of Gaussian Shading (c) TPR of PRCMark

(d) ASR of Tree-Ring (e) ASR of Gaussian Shading (f) ASR of PRCMark

Figure 6: Fine-tuning based defense dynamics across watermark schemes. We track TPR (top row) and ASR (bottom row) over fine-tuning epochs for FT, FT-SAM, and I-BAU.

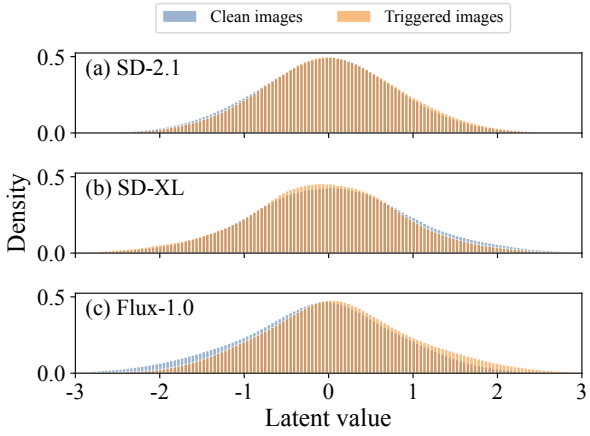


Figure 7: Latent-space stealthiness of GhostVAE. Empirical marginal density of latent values on 1,000 PRCMark images, comparing clean inputs and triggered inputs for SD-2.1, SD-XL, and FLUX-1.0.

the trigger while remaining statistically indistinguishable in latent space.

Activation Clustering [5]. We further assess whether activation clustering can separate clean and triggered inputs to GhostVAE. For this evaluation, we use 500 clean watermarked images and their corresponding triggered versions. We extract the final encoder activations of these 1,000 inputs, reduce them to lower-dimensional activations using Independent Component Analysis (ICA) [16], and then apply K-means clustering. We quantify separability using the mean optimal clustering accuracy over five runs. Specifically, because the two cluster indices produced by K-means are arbitrary, we assign them to the ground-truth clean and triggered labels in the way that yields the highest classification accuracy for each run, and report the average accuracy across five independent runs. On the SD-2.1 setting, the average optimal clustering accuracy is 52.6% and 50.1% when the ICA dimension is 256 and 512, respectively, which means that the two clusters contain nearly balanced clean and triggered samples. We further observe the same phenomenon on SD-XL, where the corresponding accuracy is 50.0% and 50.1% under the same ICA dimensions. These results indicate that activation clustering fails to reliably separate triggered inputs from clean ones.

6.4 Parameter-Space Visualization

To further probe whether GhostVAE leaves a detectable artifact footprint in the parameter space, we construct three groups of encoders, each containing 50 independently trained models: (i) benign VAE encoders obtained by fine-tuning SD-2.1 using the default training hyperparameters in Section 6.2, (ii) backdoored VAE encoders using random BadNets-style trigger [9] as described in Section 6.3, and (iii) backdoored encoders trained by GhostVAE. For each convolutional layer, we collect the corresponding weights from all encoders across three groups and jointly project them into a two-dimensional space using t-SNE [32]. Figure 8 illustrates the resulting



Figure 8: Layer-wise t-SNE visualization of convolutional weights in the VAE encoder. Each subplot (L1–L27) corresponds to a convolutional layer. We fix the LDM to SD-2.1.

layer-wise visualizations, where each point corresponds to one encoder at a specific layer. Across all layers, GhostVAE-based backdoored encoders consistently overlap with the benign cluster and remain well separated from BadNets-based backdoored encoders. These results indicate that GhostVAE does not induce a layer-specific geometric signature in the weight space, which means the introduction of the MMD-based regularization in GhostVAE effectively enhances the parameter-space stealthiness.

7 Discussion & Limitations

7.1 Regulation Suggestions

Our findings indicate that once the neural network component becomes unreliable, the effectiveness of watermarking schemes is fundamentally compromised. Consequently, we argue that regulatory efforts should place greater emphasis on the integrity of the entire model deployment pipeline underlying watermarking schemes. For example, VAEs could be required to be trained by trusted third parties, or their training processes could be subject to end-to-end regulatory oversight. Meanwhile, to prevent covert replacements, cryptographic mechanisms such as hash functions [35] should be employed to ensure the integrity of deployed VAE components.

Moreover, integrity checking alone may be insufficient if adversarial perturbations can transfer across compatible encoders. As suggested by our cross-encoder verification results in Section 6.2, replacing the verification encoder weakens but does not fully eliminate the attack. Therefore, VAE training should also consider adversarial training to improve robustness to input perturbations, while deployment pipelines should reduce the transferability of such perturbations across independently trained encoders.

We further note recent progress in text-to-image diffusion models, where image generation can be achieved without the involvement of VAEs. For instance, pMF [25] enables single-step image generation directly in the raw pixel space, without relying on a pretrained VAE or performing multi-step iterative sampling. While such architectures may be inherently resilient to our attacks, watermarking schemes specifically designed for these novel generation paradigms remain largely underexplored, thereby posing challenges even for basic provenance tracking of generated images.

7.2 Limitations

We highlight several limitations of our work. First, implanting the backdoor into the VAE encoder can affect benign watermark detection in some settings. Although GhostVAE largely preserves the original detection capability, we observe TPR degradation for certain backbone-watermark combinations, as shown in Table 4. This suggests a trade-off between attack effectiveness and function maintenance. An important next step is to design backdoor attacks that further minimize such degradation. Second, our threat model assumes that the malicious model supplier can control the encoder used during watermark detection. If the verifier instead uses an encoder that has the same architecture but different parameters, the attack can be weakened. However, with techniques that enhance adversarial transferability, such as DI^2-FGSM [46], GhostVAE can still retain a non-trivial watermark evasion ASR under cross-encoder verification. Further studying attacks in this stronger setting, and understanding how far such transferability can be improved, would be a valuable direction for future work. Third, part of our evidence for the stealthiness relies on empirical visualization. In particular, the t-SNE visualization provides only qualitative evidence and should not be interpreted as a principled detector. A longer-term goal is therefore to develop watermark evasion backdoors with cryptographic undetectability guarantees [7].

8 Conclusion

We propose GhostVAE, a stealthy VAE encoder backdoor that achieves watermark evasion under trigger activation while preserving detection behavior on benign inputs. Moreover, the universal trigger remains effective under common corruptions, purification techniques and social media pipeline, making the evasion reliable in the real world. Crucially, we show that GhostVAE is difficult to detect or mitigate using existing defenses, including a wide range of parameter-space and latent-space defense strategies. These findings underscore a fundamental gap between watermark robustness and the security of the underlying generative model, where a watermark can be robust in isolation but still fail in practice when the detection pipeline is compromised. Therefore, safeguarding semantic watermarks requires not only robust designs but also principled protection of neural network components against backdoor insertion. We hope this work motivates future research and regulatory efforts to jointly address watermark robustness and the secure deployment of generative models.

Acknowledgement

We sincerely thank the anonymous reviewers for their constructive suggestions. This work is supported by the National Cyber Security-National Science and Technology Major Project (2026ZD1500700), the Scientific Research Innovation Capability Support Project for Young Faculty (ZYGXQNJSKYCXNLZCXM-P4), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM114), the National Natural Science Foundation of China (62402273), the Guangdong Basic and Applied Basic Research Foundation (2026A1515030046), and the State Key Laboratory of Internet Architecture, Tsinghua University (HLW2025ZD14).

Ethical Considerations

Our ethical considerations consist of **stakeholder analysis**, **impact analysis**, **mitigations**, and **decision justifications**.

Stakeholder Analysis. We organize the stakeholders affected by GhostVAE according to their role in the watermark deployment and enforcement pipeline. (1) *Governance stakeholders:* This group includes regulators, auditors, platform operators, and AI image generation companies. These stakeholders care about whether deployed LDMs satisfy watermarking-related compliance requirements. (2) *Model suppliers:* They provide LDMs to generate images. Upon receiving requests from users, these models generate watermarked images, where the watermark is only detectable by model suppliers holding the watermark key. (3) *Downstream participants:* This group includes casual end users and professional content creators who use LDMs to generate images that contain watermarks. (4) *Research team:* This group includes researchers and watermark developers who study the reliability of watermarking schemes.

Impact Analysis. Our analysis identifies both positive and negative impacts on the stakeholder groups above.

Positive impacts: (1) *Enhancing Governance Auditing (for governance stakeholders):* Our work reveals a hidden risk of surface-level compliance, where a system appears compliant during auditing but can still have vulnerabilities in real deployment. Our work will help make future auditing processes of LDM watermarking systems more comprehensive. (2) *Strengthening Secure Deployment Practices (for model suppliers):* Our work helps model supplier recognize that watermark embedding alone is insufficient. They need to protect the entire detection pipeline, especially neural components like the VAE encoder, reducing the risk of hidden backdoors and improving trustworthy deployment. (3) *More Trustworthy Provenance (for downstream participants):* By exposing this hidden vulnerability, our work can help drive the development of more trustworthy provenance mechanisms, thereby reducing the risk that downstream participants rely on provenance signals that fail in practice. (4) *Advancing Research on Watermark Security (for research team):* Our work exposes a previously overlooked attack surface in semantic watermarking systems, motivating future research on verifier integrity, backdoor-resistant watermarking, and end-to-end security for LDM deployment pipelines.

Negative impacts: (1) *Reduced Confidence in Watermark-Based Enforcement (for governance stakeholders):* Our work may weaken confidence in current watermark-based enforcement by showing that these systems can be less reliable than expected. (2) *Increased Misuse and Commercial Risks (for model suppliers):* Our work may help malicious suppliers selectively disable watermark detection, while also creating commercial risks for benign providers, such as customer churn. (3) *Harder-to-Trace Harmful Content (for downstream participants):* Our work may enable malicious users to generate images that evade watermark detection, making harmful or misleading content harder to trace. (4) *Challenging Existing Research Assumptions (for research team):* Our work may challenge reliability assumptions in existing semantic watermarking schemes by showing that watermarked latents alone are insufficient for security evaluation. While this may prompt researchers to revisit prior protocols and claims, our findings do not invalidate the underlying cryptographic assumptions; instead, they expose risks arising from the integration of cryptographic primitives with AI components.

Mitigations. We acknowledge that this work may be misused by malicious actors to better understand or reproduce watermark evasion attacks. For example, a malicious supplier–user coalition could use a backdoored watermarking pipeline to generate and disseminate harmful synthetic images while evading traceability. To mitigate such risks, we adopt the following measures.

As discussed in Section 7.1, our mitigation suggestions include trusted third-party VAE training, end-to-end oversight of training and deployment, integrity checks for deployed VAE components, and designing more secure watermarking schemes. We emphasize that any use of our findings should remain within applicable law and regulation.

Meanwhile, to mitigate real-world threats, we did not release any attack-ready artifacts generated in this work, such as optimized triggers, backdoored VAE encoders, or directly reusable checkpoints. Our evaluation was restricted to controlled offline experiments on local systems, and we did not test on deployed large-scale services, engage with real users, or attempt any real-world watermark evasion or misuse. These restrictions were intended to limit operational abuse while preserving the value of the work as a security assessment.

Decision Justifications.

Decision to Conduct: It was appropriate to conduct this work because the vulnerability we studied is a practical deployment-level risk in semantic watermarking systems whenever detection relies on neural components that can be backdoored. Our motivation was to assess whether this attack is practically feasible and whether it can remain stealthy in deployment. Before starting this work, we explicitly considered the relevant stakeholders, the misuse risks, and the conditions under which the work could be responsibly conducted. We also carried out all experiments in a controlled offline setting.

Decision to Publish: Although backdoor research on general AI models has been extensively published, its integration with watermarking scenarios has not yet been well studied. Without disclosure, this vulnerability could remain hidden and sustain misplaced confidence in watermark-based enforcement. After weighing the misuse risks against the defensive value of disclosure, we concluded that publication is justified, as it can help the community avoid false assurance, motivate more secure watermarking schemes, and strengthen the auditing process. We believe the benefits of publication outweigh the risks, and by sharing our findings through a high-impact platform like USENIX, we can promote long-term security of the watermarking ecosystem.

Open Science

To support reproducibility, we release an artifact package that contains (i) the necessary configuration files, (ii) all defense and evaluation scripts used in our study, and (iii) the full implementation of GhostVAE. Specifically, the repository provides an integrated workflow, including environment setup, dataset preparation, universal-trigger optimization, backdoored VAE-encoder training, and representative defense runs.

In particular, we provide two standard dependency specifications, `requirements_dif.txt` and `requirements_ghostvae.txt`, for different task stages, together with example commands. The main results in this paper can be reproduced by following the instructions in `README.md`. All artifacts are available via https://github.com/CryptoAILab/GhostVAE and https://doi.org/10.5281/zenodo.20391190.

References

[1] Stability AI. sd-vae-ft-ema. https://huggingface.co/stabilityai/sd-vae-ft-ema, 2023.

[2] Stability AI. sd-vae-ft-mse. https://huggingface.co/stabilityai/sd-vae-ft-mse, 2023.

[3] Stability AI. sdxl-vae. https://huggingface.co/stabilityai/sdxl-vae, 2023.

[4] Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. WAVES: benchmarking the robustness of image watermarks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.

[5] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019*, CEUR Workshop Proceedings, 2019.

[6] Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In *Annual International Cryptology Conference*, pages 325–347. Springer, 2024.

[7] Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models : [extended abstract]. *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 931–942, 2022.

[8] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(null):723–773, March 2012.

[9] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[10] Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

[11] Wenbo Guo, Lun Wang, Yan Xu, Xinyu Xing, Min Du, and Dawn Song. Towards inspecting and eliminating trojan backdoors in deep neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 162–171, 2020.

[12] Gustavosta. Stable-diffusion-prompts. URL: https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[14] J. L. Hodges. The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3:469–486, 1958.

[15] Hugging Face Inc. Hugging face hub. https://huggingface.co, 2024.

[16] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4):411–430, 2000.

[17] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 2043–2059. IEEE, 2022.

[18] Jinsu Kim, Yunhun Nam, Minseon Kim, Sangpil Kim, and Jongheon Jeong. Blurguard: A simple approach for robustifying image protection against AI-powered editing. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[19] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[20] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

[21] Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. Reconstructive neuron pruning for backdoor defense. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19837–19854. PMLR, 2023.

[22] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings*, volume 11050 of *Lecture Notes in Computer Science*, pages 273–294. Springer, 2018.

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[24] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[25] Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows. *arXiv preprint arXiv:2601.22158*, 2026.

[26] Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao Chen, Yunhao Zhao, Hanxun Huang, Yige Li, Yutao Wu, Jiaming Zhang, Xiang Zheng, Yang Bai, Yiming Li, Zuxuan Wu, Xipeng Qiu, Jingfeng Zhang, Xudong Han, Haonan Li, Jun Sun, Cong Wang, Jindong Gu, Baoyuan Wu, Siheng Chen, Tianwei Zhang, Yang Liu, Mingming Gong, Tongliang Liu, Shirui Pan, Cihang Xie, Tianyu Pang, Yinpeng Dong, Ruoxi Jia, Yang Zhang, Shiqing Ma, Xiangyu Zhang, Neil Gong, Chaowei Xiao, Sarah M. Erfani, Tim Baldwin, Bo Li, Masashi Sugiyama, Dacheng Tao, James Bailey, and Yu-Gang Jiang. Safety at scale: A comprehensive survey of large model and agent safety. *Found. Trends Priv. Secur.*, 8(3-4):254–469, 2025.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[28] Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[29] Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 20937–20946. Computer Vision Foundation / IEEE, 2025.

[30] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16805–16827. PMLR, 2022.

[31] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[32] Pavlin G. Policar, Martin Strazar, and Blaz Zupan. opentsne: A modular *Python* library for t-sne dimensionality reduction and embedding. *J. Stat. Softw.*, 109(3), 2024.

[33] Nate Raymond. Us judicial panel to develop rules to address ai-produced evidence. https://www.reuters.com/legal/transactional/us-judicial-panel-develop-rules-address-ai-produced-evidence-2024-11-08/, November 2024.

[34] Reuters Fact Check. Fact check: Images of aircraft landings into flaming beirut airport are ai-generated. https://www.reuters.com/fact-check/images-aircraft-landings-into-flaming-beirut-airport-are-ai-generated-2024-10-29/, October 2024.

[35] Ronald L. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, April 1992. URL: https://www.rfc-editor.org/info/rfc1321.

[36] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021.

[37] Artur Soboń and Sylwia Stachowiak. Chacha20 cipher cryptanalysis through sat problem solving. In *2024 IEEE 17th International Scientific Conference on Informatics (Informatics)*, pages 355–361, 2024.

[38] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[39] The European Parliament and of the Council. Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858, (eu) 2018/1139 and (eu) 2019/2144 and directives 2014/90/eu, (eu) 2016/797 and (eu) 2020/1828 (artificial intelligence act). Official Journal of the European Union, July 2024. URL: https://eurlex.europa.eu/eli/reg/2024/1689/oj/eng.

[40] Jordan Vice, Naveed Akhtar, Richard I. Hartley, and Ajmal Mian. BAGM: A backdoor attack for manipulating text-to-image generative models. *IEEE Trans. Inf. Forensics Secur.*, 19:4865–4880, 2024.

[41] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019.

[42] Haonan Wang, Qianli Shen, Yao Tong, Yang Zhang, and Kenji Kawaguchi. The stronger the diffusion model, the easier the backdoor: Data poisoning to induce copyright breaches without adjusting finetuning pipeline. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning - The Good, the Bad, and the Ugly*, 2024.

[43] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Advances in Neural Information Processing Systems*, volume 36, pages 58047–58063. Curran Associates, Inc., 2023.

[44] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, Mingli Zhu, Ruotong Wang, Li Liu, and Chao Shen. Backdoorbench: A comprehensive benchmark and analysis of backdoor learning. *Int. J. Comput. Vis.*, 133(8):5700–5787, 2025.

[45] X. Formerly twitter. URL: https://x.com.

[46] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2725–2734, 2019.

[47] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A. Gunter, and Bo Li. Detecting AI trojans using meta neural analysis. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 103–120. IEEE, 2021.

[48] Xiaoyun Xu, Zhuoran Liu, Stefanos Koffas, and Stjepan Picek. Towards backdoor stealthiness in model parameter space. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, CCS 2025, Taipei, Taiwan, October 13-17, 2025*, pages 2863–2876. ACM, 2025.

[49] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[50] Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Wei Ming Zhang, and Neng H. Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12162–12171, 2024.

[51] Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[52] Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, pages 1577–1587. ACM, 2023.

[53] Shengfang Zhai, Jiajun Li, Yue Liu, Huanran Chen, Zhihua Tian, Wenjie Qu, Qingni Shen, Ruoxi Jia, Yinpeng Dong, and Jiaheng Zhang. Efficient input-level backdoor defense on text-to-image synthesis via neuron activation variation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15182–15193, October 2025.

[54] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative AI. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

[55] Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ODE solver with empirical model statistics. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[56] Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part V*, volume 13665 of *Lecture Notes in Computer Science*, pages 175–191. Springer, 2022.

[57] Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4443–4454. IEEE, 2023.

A RAPSD Calculation

Given an image x and its triggered counterpart $x + \mathcal{G}(\delta;\sigma)$, we summarize their frequency content using the radially-averaged power spectral density (RAPSD). For an input image x, we compute its Fourier frequency matrix $f := F(x) \in \mathbb{C}^{h\times w}$, where h and w denote the height and width of the Fourier grid (matching the spatial resolution of x), and (u,v) indexes a discrete frequency coordinate on this $h \times w$ grid. We then partition the coordinates of f into B radial bands according to their distance to the spectrum center, yielding B disjoint sets $\{\mathcal{B}_b\}_{b=1}^{B}$, where $\mathcal{B}_b$ contains the frequency coordinates that fall into the b-th band and $|\mathcal{B}_b|$ denotes its cardinality. RAPSD is defined as a B-dimensional vector whose b-th entry is the average squared magnitude of FFT coefficients within $\mathcal{B}_b$:

$$\mathrm{RAPSD}_b(f) := \frac{1}{|\mathcal{B}_b|} \sum_{(u,v)\in\mathcal{B}_b} |f(u,v)|^2. \tag{1}$$

Collecting all B band-wise entries yields the RAPSD vector.

B Experimental Details

B.1 Watermark Detection Threshold Setting

We summarize the detection thresholds for each semantic watermarking on each backbone model. These thresholds shown in Table 11 are fixed across all experiments in our evaluation. Different models and watermark designs induce different calibration characteristics; thus, a single global threshold would conflate detection stringency across settings and lead to unfair comparisons. We use the official thresholds when they are specified in the original watermarking papers; for other backbone-watermark combinations, we calibrate thresholds to achieve high TPR and low FPR on benign models. We further assess threshold sensitivity on Gaussian Shading with SD-XL by varying the threshold from 0.67 to 0.60 and 0.70. TPR/FPR/ASR change to 73.0%/0.0%/100.0% and

Table 11: Detection threshold configurations for watermark verification. For PRCMark, we show the σ_{prc} here.

Scheme	Threshold		
	SD-2.1	SD-XL	Flux-1.0
Tree-Ring	0.59	0.67	0.60
Gaussian Shading	1.00	0.98	0.95
PRCMark	0.15	0.15	0.15

Table 12: Hyperparameter settings for parameter-level backdoor defenses.

Defense	Hyperparameters
CLSP	pruning rate = 0.15%, prune steps = 20
CLP	sweep pruning threshold from 0.5 to 5.0 with step size 0.5
ANP	epochs=4, batch size=2, ε = 0.4, lr_{inner} = 0.4, lr_{outer} = 0.2, sweep pruning threshold from 0.1 to 0.9 with step size 0.1
FT	epoch = 10, $lr = 1e-5$
FT-SAM	epoch = 10, $lr = 1e-5$, $\rho_{\max} = 2.0$, $\rho_{\min} = 0.05$, $\rho_{schedule} = cosine$
I-BAU	epoch = 10, $lr = 1e-5$, n_{rounds} = 4, K = 5, pert steps = 50, unlearn portion = 0.01, unlearn batches = 300
KS test	significance level $\alpha = 0.05$

99.6%/1.3%/92.5%, respectively. While these changes reflect the expected threshold trade-off, GhostVAE consistently maintains strong evasion performance, indicating that our findings are robust to reasonable threshold choices.

B.2 Defense Methods Parameter Setting

We disclose the hyperparameter settings for the model pruning and fine-tuning defenses evaluated in Section 6.2. Table 12 lists the exact configurations used in our experiments. Compared to the default hyperparameter configuration, we reduce the number of epochs because training the VAE encoder is computationally time-consuming.

C Sensitivity to Step Budgets for Watermark Removal Attacks

We perform a sensitivity study on the step budgets of two iterative watermark removal baselines: the optimization steps T_{imprint} used by Imprint and the denoising steps T_{reg} used by Regeneration. We sweep each budget while holding all other settings fixed, and report the ASR and image quality.

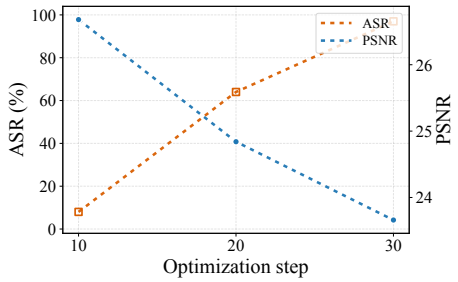


Figure 9: Sensitivity of the Imprint attack to the optimization steps T_{imprint}. We report the resulting ASR (left axis) and image quality measured by PSNR (right axis) as T_{imprint} varies.

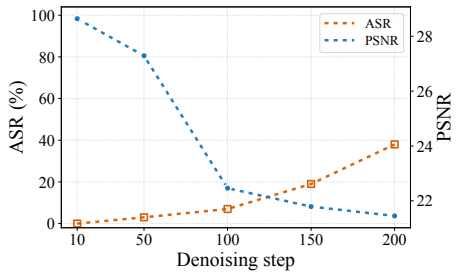


Figure 10: Sensitivity of the Regeneration attack to the denoising steps T_{reg}. We report the resulting ASR (left axis) and image quality measured by PSNR (right axis) as T_{reg} varies.

Both iterative baselines exhibit a clear trade-off: increasing the step budget improves watermark evasion but degrades image quality and increases runtime. For Imprint attack results shown in Figure 9, raising the optimization steps from T_{imprint}=10 to 30 sharply increases the ASR from 3% to 97%, but this gain comes with a substantial PSNR drop, indicating noticeably stronger visual distortion. For Regeneration, Figure 10 shows the same pattern: larger denoising budgets increase ASR monotonically, yet PSNR steadily deteriorates, so the attack becomes more destructive and costly as it becomes more effective. These results motivate our main-paper choices (T_{imprint}=30 and T_{reg}=200), which place both baselines in their stronger regimes while making the associated quality and efficiency costs explicit.