PGID: Progressive Guided Inversion and Denoising for Robust Watermark Detection

Minh Quoc Duong **Chun Tong Lei** **Chun Pong Lau***
City University of Hong Kong
{mduong2-c, ctlei2-c}@my.cityu.edu.hk, cplau27@cityu.edu.hk

Abstract

With the proliferation of AI-generated images, digital watermarking has become an essential safeguard for protecting intellectual property and mitigating malicious exploitation. Recent works on semantic watermarking have enabled efficient copyright protection for diffusion models. However, the dependence of semantic watermarking on diffusion inversion for watermark detection creates a critical vulnerability. Imprint removal and forgery attacks exploit this weakness to produce deceptive results. Our analysis reveals that these attacks succeed by displacing watermarked latents into the *unwatermarked region*, while guiding unwatermarked latents into the *watermarked region*. Based on that, we propose **Progressive Guided Inversion and Denoising (PGID)**, the first plug-and-play, training-free noise extraction framework designed to defend against both attack strategies. PGID effectively defends by projecting perturbed latents back to the region where they originally belong. The projection is achieved by eliminating intermediate latent deflections and mitigating adversarial perturbations through progressive inversion-denoising cycles. Comprehensive evaluations across multiple schemes demonstrate that PGID successfully restores detection reliability by recovering removed watermarks and identifying forged instances.

1 Introduction

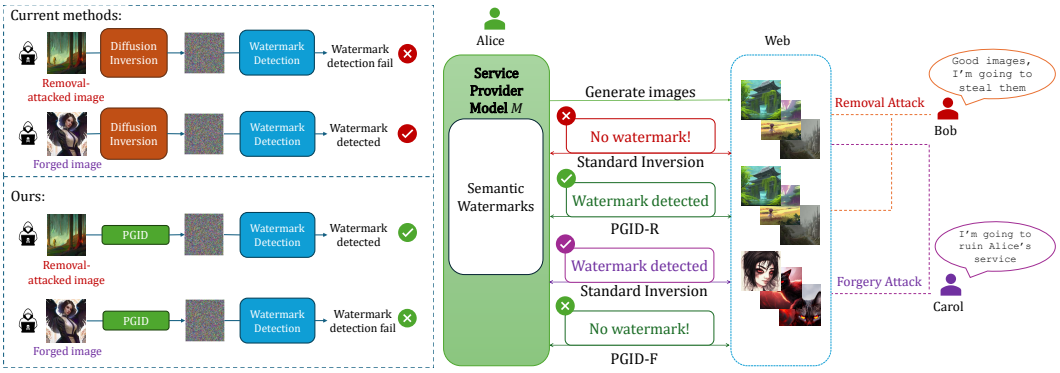
Recent advances in diffusion models [1, 2, 3] have empowered users to generate high-fidelity images at an unprecedented scale effortlessly, leading to a surge in AI-generated digital content. However, with this massive growth comes an increasing range of risks. The malicious use of generative models to spread disinformation and misguide communities presents a profound societal threat. Furthermore, without safeguards, service providers become vulnerable to intellectual property theft, unauthorized exploitation, and severe reputational damage.

Watermarking is an effective solution to mitigate the above risks by enabling AI-generated content detection, copyright authentication, and provenance tracing. In recent years, semantic watermarks [4, 5, 6, 7, 8] emerged as a state-of-the-art watermarking paradigm for diffusion models. These methods function by injecting watermark signals into the initial noise of the diffusion process and use it as input for image generation. Watermark extraction is achieved by tracing the generative process backward through diffusion inversion to retrieve an estimate of the initial noise. This recovered noise is subsequently decoded for watermark detection.

Even though different semantic watermark schemes have different ways of injecting and decoding the embedded signals, they all share the same dependence on an accurate inversion function for initial noise retrieval, which is typically DDIM [9]. Exploiting this dependency, Müller et al. (2025)[8] have demonstrated the vulnerability of this watermarking paradigm to black-box imprint forgery and removal attacks. Consequently, attackers can either falsely attribute illicit content to service

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) (b)

Figure 1: Overview of the PGID framework and threat model. (a) Current semantic watermark methods are vulnerable to imprint forgery and removal attacks. PGID can recover watermark signals and reject forged images, providing effective defense. (b) The application scenario for PGID.

providers through forgery or bypass accountability measures entirely by removing watermarks. This highlights a need for more robust extraction frameworks to complement current one-stage inversion approaches [10, 11].

We first observe that for any fixed watermark signal in an initial watermarked noise z_T^*, the latent space is partitioned into a *watermarked region* and an *unwatermarked region*. Under this formulation, removal attacks displace watermarked latents into the unwatermarked region, while forgery attacks guide cover images into the watermarked region. Based on that observation, we propose **Progressive Guided Inversion and Denoising (PGID)**, the first plug-and-play training-free noise extraction framework for defense against both imprint forgery and removal attacks. Our results show that, under the proposed unified framework, one can: (1) recover watermarks from images subjected to removal attack, and (2) differentiate between forged and authentic watermarked images. In our work, we refer to the usage of PGID against the removal attack and forgery attack as PGID-R and PGID-F, respectively.

Fundamentally, our method projects the attacked latent back to the region it originally was in. This is achieved from the observation that imprint attacks induce cumulative distributional bias at every intermediate timestep. Our framework eliminates these deflections by traversing the diffusion trajectory through asymmetric cycles where **we denoise more than we invert** at each cycle, effectively mitigating the introduced perturbations while projecting the latent representation back toward its intended manifold.

In summary, our contributions in this paper are:

- We formalize the concepts of watermarked and unwatermarked regions for an initial noise watermark signal and empirically demonstrate that imprint attacks displace the latent representation out of its respective region to disrupt detection.
- From there, we propose **Progressive Guided Inversion and Denoising (PGID)**, the first plug-and-play training-free noise extraction framework unifying both imprint forgery and removal defense for semantic watermarks.
- We validate the effectiveness of PGID with cutting-edge semantic watermarking schemes. Empirical results demonstrate that PGID is effective across different watermarking schemes and different attack settings. In the removal case, PGID can reliably recover the obscured watermark. To the best of our knowledge, it is the first framework capable of such a restoration.

2 Related Works

2.1 Image Watermarking for Diffusion Models

Image watermarking for diffusion models can be categorized into two primary methodologies: **post-processing** and **in-processing**. Post-processing methods apply watermarks directly to the final image

using traditional transform-domain techniques [12] or deep learning-based encoders [13, 14, 15]. However, these methods often compromise image fidelity, do not guarantee undetectability, and offer limited robustness against removal [16, 17]. In-processing methods can be further divided into two subtypes: fine-tuning schemes and semantic watermarking schemes. Fine-tuning methods represent a more integrated approach where the watermark is embedded directly into the model's parameters, typically by fine-tuning the Variational Autoencoder (VAE) decoder [18] or the noise predictor [19]. While this ensures all outputs are traceable and maintains high image quality, it incurs significant computational costs. Semantic watermarks remove the need for extensive model retraining by manipulating the initial input noise to a diffusion model and recovering the watermark by inverting the diffusion process. Techniques such as Tree Ring [4] and RingID [6] utilize Fourier-space structuring to ensure geometric invariance, while methods like Gaussian Shading [5] and PRC [20] bias the initial latent distribution to achieve efficient watermarking. In our work, we specifically focus on this semantic watermarking paradigm.

2.2 Watermark Forgery and Removal

As watermarking techniques advance, a parallel field of research has emerged focusing on the vulnerabilities of these systems to removal and forgery. The removal of post-processing image watermarks has been extensively investigated through various techniques, ranging from traditional photometric and degradation distortions [21, 22] to more sophisticated deep learning-based methods [17, 23]. Furthermore, recent advancements have introduced regeneration and reconstruction approaches [24, 25] that leverage diffusion-based purification [26, 27]. Likewise, forgery attacks on these traditional schemes are also well studied, including training a surrogate generative model or a preference model to estimate the watermark [28, 29]. Most semantic watermark strategies have shown strong robustness against these adversaries [30, 7, 31]. However, recently proposed imprint forgery and removal attacks [8] have highlighted a security gap in the paradigm's dependence on diffusion inversion. There have been works attempting to improve robustness against forgery attacks by binding the watermarks to image semantics [7, 32], but these remain susceptible to removal. Our method fills the gap by tackling both removal and forgery attacks simultaneously within a unified framework.

3 Preliminaries

3.1 Latent Diffusion Models (LDMs) and DDIM

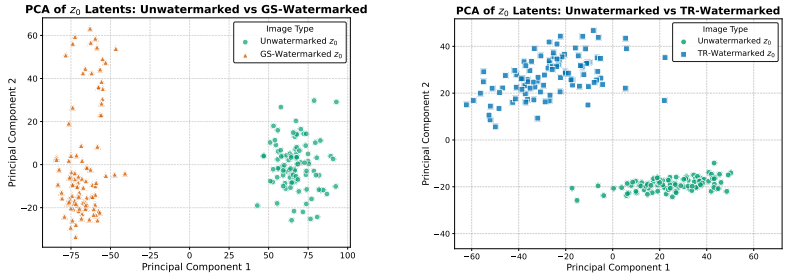
Diffusion models [33, 34, 35, 36, 9] are generative models that are defined by a *forward process* and a parameterized *denoising process*. LDMs [1] are diffusion models that operate in a compressed latent space rather than the pixel domain to achieve high-fidelity generation with computational efficiency. Given an image $x \sim q(x)$, an LDM first maps x to a latent representation $z_0 = \mathcal{E}(x)$ via an encoder $\mathcal{E}$. Then the *forward process* progressively degrades this latent z_0 by adding Gaussian noise to generate a sequence of noisy samples $\{z_t\}_{t=1}^{T}$. At any given timestep $t \in [1, T]$, the noisy latent z_t is defined as a linear combination of the initial latent and standard Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon \tag{1}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}(1 - \beta_i)$ is the cumulative product of a predefined variance schedule $\{\beta_t\}_{t=1}^{T} \in (0, 1)$.

The *denoising process* aims to reverse this degradation to generate new data. It starts by sampling z_T from the standard Gaussian distribution $p(z_T) = \mathcal{N}(z_T; 0, I)$. A parameterized neural network $\epsilon_\theta(z_t, t)$ is trained to estimate the added noise ϵ that was added at timestep t. A sampler then uses $\epsilon_\theta(z_t, t)$ to compute a less noisy latent z_{t-1} from z_t. Numerous samplers have been proposed, and Denoising Diffusion Implicit Models (DDIM) [9] is among the most common methods because of its fast sampling speed and deterministic nature. With deterministic DDIM, a sampling step from t to $t - 1$ is defined as:

$$z_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left(\frac{z_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(z_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon_\theta(z_t, t) \tag{2}$$



(a) (b)

Figure 2: Visualization of latent manifold separation. PCA projections of the latent space $\mathcal{Z}_0$ for (a) Gaussian Shading and (b) Tree Ring. The visualization demonstrates a clear structural differentiation between the watermarked region and the unwatermarked region for a fixed initial noise signal.

An image is then generated by iteratively applying this deterministic update from $t = T$ down to $t = 1$ to obtain the clean latent z_0, which is subsequently mapped back into the pixel space using a decoder $\mathcal{D}$, such that the final image is $x' = \mathcal{D}(z_0)$.

3.2 Inverse DDIM

Because DDIM is deterministic, one can follow the trajectory backwards from timestep 0 up to T to estimate the initial noise z_T. Given a generated image x', we can first map it back to the latent domain $z_0 = \mathcal{E}(x')$. The Inverse DDIM process steps forward in time from t to $t + 1$ as follows:

$$z_{t+1} = \sqrt{\bar{\alpha}_{t+1}} \left(\frac{z_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(z_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t+1}} \epsilon_\theta(z_t, t) \tag{3}$$

For our work, we denote a single step of DDIM sampling (from timestep t to $t - 1$) and inverse DDIM (from timestep t to $t + 1$) as $Denoise(z, t)$ and $Inverse(z, t)$, respectively. Let $\mathcal{T}_{0\to T}(z_0)$ represent the inversion process that yields the latent trajectory $\{z_t\}_{t=0}^{T}$, and $I_{0\to T}(z_0)$ denote the mapping to the final estimated noise z_T.

4 Method

In this section, we describe how service providers can utilize PGID to protect themselves from imprint forgery and removal attacks. Subsequently, we formalize the latent-space partition into watermarked and unwatermarked regions given a watermark signal, and describe the corresponding behavior of removal and forgery attacks in this space. Built upon that, we then present PGID in detail.

4.1 Application Scenarios

Scenarios. We consider a scenario in which Alice provides an API for users to access her generative model M. The model M is kept hidden and closed-source from users. Alice employs a semantic watermarking method for copyright protection and traceability. However, Alice faces threats from two types of malicious users: Bob, who seeks to steal Alice's generated images by performing an imprint removal attack to claim them as his own, and Carol, who seeks to damage Alice's reputation by using an imprint forgery attack to embed Alice's watermark into unauthorized or illicit images. If Alice relies solely on a standard extraction pipeline, both attacks succeed: her watermark is not detected in Bob's stolen images, whereas it is falsely detected in Carol's forged images. Consequently, Alice is left vulnerable to copyright loss and potential legal liability.

Usage. Given an arbitrary image, Alice does not know whether it has undergone a removal attack, a forgery attack, or no attack at all. To counteract both threats, Alice integrates PGID, which requires only her model M, into her system. For any suspect image, Alice first runs the standard watermark extraction, using its outcome to dictate her next step. *(1) Verifying Rejections (Defending against Bob):* If the standard detection yields a *negative* result, but Alice suspects the image was stolen from her service, she applies PGID-R. If PGID-R successfully recovers the hidden watermark, she confirms a removal attack occurred and reclaims her copyright. *(2) Verifying Authenticity (Defending*

Algorithm 1 PGID

Require: Suspect image x_{atk}, stopping timestep k, number of skip inversion steps s, guidance strength γ, and model M used for watermarked image generation with encoder $\mathcal{E}$ and decoder $\mathcal{D}$.

Stage I: Trajectory Pre-computation

$z_0^{atk} \leftarrow \mathcal{E}(x^{atk})$

Let $\{z_t^{atk}\}_{t=0}^{T} \leftarrow \mathcal{T}_{0\to T}(z_0^{atk})$ ▷ Pre-compute the standard inversion trajectory

Stage II: Progressive Latent Refinement

$\hat{z} \leftarrow z_0^{atk}$ ▷ Initialization

for $i = 1$ to k **do**

  for $t = 1$ to $i - s$ **do** ▷ Invert $i - s$ steps

    $\hat{z} \leftarrow Inverse(\hat{z}, t-1)$

    $\hat{z} \leftarrow \hat{z} + \gamma(\hat{z} - z_t^{atk})$

  end for

  for $t = i$ down to 1 **do** ▷ Denoise i steps

    $\hat{z} \leftarrow Denoise(\hat{z}, t)$

  end for

end for

$z_0 \leftarrow \hat{z}$ ▷ Finish refinement

Stage III: Final Inversion

$z_T \leftarrow I_{0\to T}(z_0)$ ▷ Final full inversion

return z_T

against Carol): If the standard detection yields a *positive* result on a suspicious or illicit image, she applies PGID-F. If the watermark is no longer detected, she confirms that it was a forged instance. The application scenario is presented in Figure 1b.

4.2 Watermarked and Unwatermarked Latent Regions

Formalization. Jain et al. (2025)[37] first proposed the existence of a *watermarked region* for forgery exploitation. In our work, we further formalize and empirically show that such a region and its complement, the *unwatermarked region*, exist with respect to the specific watermark signal embedded in an **initial watermarked noise** z_T^*. Given a watermarking method and an initial watermarked noise z_T^*, let $\mathcal{D}_{z_T^*} : \mathcal{Z}_T \to \{0, 1\}$ be a binary detection function that evaluates whether a given noise vector contains the specific watermark signal in z_T^*. We formalize the watermarked region $\mathcal{R}_w(z_T^*)$ as the set of all latents z_0 that invert to a noise vector triggering a successful detection:

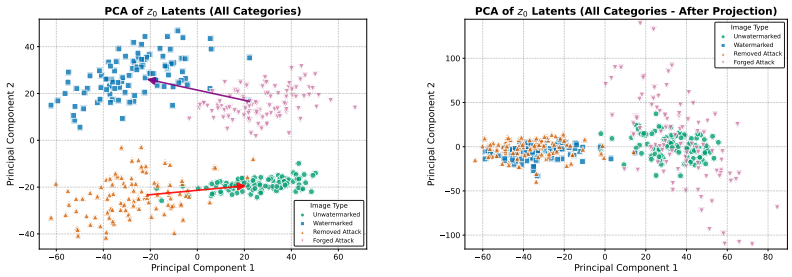
$$\mathcal{R}_w(z_T^*) = \{z_0 \in \mathcal{Z}_0 \mid \mathcal{D}_{z_T^*}(I_{0\to T}(z_0)) = 1\}$$

In other words, there exist many z_0 mapping to a noise z_T through the inversion process that contains the specific watermark signal from z_T^*, which collectively constitute the watermarked region $\mathcal{R}_w(z_T^*)$. Conversely, we define the *unwatermarked region* $\mathcal{R}_u(z_T^*)$ as the complement of the watermarked region in the latent space. It comprises all latents z_0 whose inverted noise fails to yield a detectable signal for z_T^*:

$$\mathcal{R}_u(z_T^*) = \{z_0 \in \mathcal{Z}_0 \mid \mathcal{D}_{z_T^*}(I_{0\to T}(z_0)) = 0\}$$

Equivalently, $\mathcal{R}_u(z_T^*) = \mathcal{Z}_0 \setminus \mathcal{R}_w(z_T^*)$. We empirically show that these regions exist by performing a toy experiment on Tree Ring and Gaussian Shading, where we inject only a single watermark signal across watermarked images. For Tree Ring, we inject only one key pattern across all images (this is the default mechanism of Tree Ring). For Gaussian Shading, we generate all images using a single key and message. Consequently, if we fix the initial noise of any single watermarked image as z_T^*, the latents z_0 of the remaining watermarked images belong to the corresponding watermarked region $\mathcal{R}_w(z_T^*)$. Under this setup, we perform PCA on 100 watermarked images and 100 unwatermarked images from the two schemes. From Figure 2, the watermarked and unwatermarked regions are clearly separated when conditioned on a specific watermark signal, confirming our hypothesis.

Threat Model: Removal and Forgery. Let $x^{(w)}$ and $x^{(c)}$ be a watermarked image generated from initial watermarked noise z_T^* and an unwatermarked cover image, respectively. Let $z_0^{(w)} = \mathcal{E}(x^{(w)})$ and $z_0^{(c)} = \mathcal{E}(x^{(c)})$ where $\mathcal{E}$ is the encoder from the service provider's model M. Under this formulation, an imprint removal attack can be viewed as finding a perturbation δ that shifts



(a) (b)

Figure 3: (a) PCA visualization of latent-space clean and attacked regions for Tree Ring. Forgery guides cover latents toward the watermarked region, while removal attacks displace watermarked latents into the unwatermarked region. (b) PCA visualization of latent-space clean and attacked regions for Tree Ring after projection by PGID. It can be observed that attacked latents are projected back to their original regions.

an originally watermarked latent into the unwatermarked region ($z_0^{atk} = z_0^{(w)} + \delta \in \mathcal{R}_u(z_T^*)$). Conversely, an imprint forgery attack constructs a perturbation to force a cover latent into the watermarked region ($z_0^{atk} = z_0^{(c)} + \delta \in \mathcal{R}_w(z_T^*)$). We show this by visualizing PCA of watermarked, unwatermarked, removal-attacked, and forged latents from Tree Ring. As illustrated in Figure 3a, forged latents converge toward the watermarked region, while removal-attacked latents are displaced into the unwatermarked distribution.

4.3 Progressive Guided Inversion and Denoising (PGID)

From the above results, a valid defense strategy against the removal attack is to project z_0^{atk} back to an instance in the watermarked region, $z_0 \in \mathcal{R}_w(z_T^*)$. Likewise, against forgery, we can project z_0^{atk} back to the unwatermarked region $z_0 \in \mathcal{R}_u(z_T^*)$. This intuition is achieved by our PGID method.

Overview. PGID can be divided into three stages as described in Algorithm 1. The first stage involves running the standard DDIM inversion to obtain attacked intermediate latents $\{z_t^{atk}\}_{t=0}^T$. The second stage is where we perform progressive latent refinement and obtain the projected z_0 by utilizing the precomputed intermediate latents from the first stage. The final stage is running standard inversion with the projected z_0 to obtain the final noise z_T.

Progressive Latent Refinement. We focus on the second stage of our framework, where we progressively refine our latent projection. As described in Figure 4, we found that the final inversion error caused by the attacks is cascaded through intermediate timesteps. To fix the errors caused by this, we guide the inversion trajectory of the projected latent away from the precomputed $\{z_t^{atk}\}_{t=0}^T$ during inversion. Then, by denoising back, using the model's clean manifold projection capability, the latent has its inversion trajectory refined. To further remove the effects of the perturbation, we skip some inversion steps while keeping the same denoising steps every cycle. The intuition behind this process is to treat the introduced adversarial perturbation as additional noise on the original clean latent. Therefore, by inverting fewer steps than denoising, we can project the perturbed latent to the clean manifold more effectively. This stage uses three hyperparameters: the stopping timestep $k < T$, the number of skip inversions s, and guidance strength γ. The stopping timestep k confines our progressive cycle to operate on the first k timesteps, s determines how many inversion steps we skip in each cycle, while γ dictates the intensity of trajectory guidance. With different sets of these hyperparameters, PGID can be used to effectively defend against both removal and forgery. Figure 3b shows that PGID successfully projects attacked latents back to their original regions: removal-attacked back to watermarked and forged back to unwatermarked.

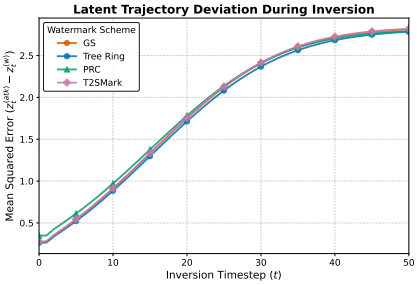


Figure 4: Visualization of the trajectory deviation MSE caused by the removal attack from $t = 0$ to $t = 50$. The inversion discrepancy accumulates across intermediate timesteps.

Table 1: Results against imprint removal attack for four semantic watermarking schemes and their PGID-augmented variants. Results are reported for the attack at 50/100/150 optimization steps. Higher AUC and Det. indicate higher robustness to the attack. The best result for each scheme is in **bold**.

Method	Attacker Model: SD 2.1			Attacker Model: SD 1.5		
	Det.↑	Bit Acc.↑	AUC↑	Det.↑	Bit Acc.↑	AUC↑
TR	0.24/0.13/0.11	—	0.4085/0.1781/0.1363	0.25/0.13/0.10	—	0.4235/0.1942/0.1393
TR+P	**0.99/1.00/1.00**	—	**0.9998/0.9988/0.9998**	**1.00/1.00/1.00**	—	**1.0000/1.0000/0.9999**
GS	0.00/0.00/0.00	0.2582/0.0342/0.0077	0.0747/0.0000/0.0000	0.01/0.00/0.00	0.2891/0.0454/0.0114	0.1022/0.0000/0.0000
GS+P	**1.00/1.00/1.00**	**0.9975/0.9987/0.9981**	**1.0000/1.0000/1.0000**	**1.00/1.00/1.00**	**0.9964/0.9981/0.9968**	**1.0000/1.0000/1.0000**
PRC	0.00/0.00/0.00	0.5012/0.5038/0.5020	—	0.00/0.00/0.00	0.5011/0.4980/0.5039	—
PRC+P	**0.97/1.00/1.00**	**0.9590/1.0000/0.9847**	—	**0.97/1.00/0.98**	**0.8907/0.9841/0.9710**	—
T2SMark	0.50/0.94/**1.00**	0.5093/0.4977/0.4965	0.8695/0.9991/**1.0000**	0.44/0.94/**1.00**	0.5177/0.4964/0.4946	0.8806/0.9976/**1.0000**
T2SMark+P	**1.00/1.00/1.00**	**0.9999/0.9999/1.0000**	**1.0000/1.0000/1.0000**	**1.00/1.00/1.00**	**0.9998/0.9999/1.0000**	**1.0000/1.0000/1.0000**

Table 2: Results against imprint forgery attack on MS-COCO dataset for four semantic watermarking schemes and their PGID-augmented variants. Results are reported for the attack at 50/100/150 optimization steps. Lower AUC and Det. indicate higher robustness to the attack. The best result for each scheme is in **bold**.

Method	Attacker Model: SD 2.1			Attacker Model: SD 1.5		
	Det.↓	Bit Acc.↓	AUC↓	Det.↓	Bit Acc.↓	AUC↓
TR	1.00/1.00/1.00	—	1.0000/1.0000/1.0000	1.00/1.00/1.00	—	1.0000/1.0000/1.0000
TR+P	**0.02/0.04/0.07**	—	**0.5472/0.6048/0.6728**	**0.04/0.08/0.09**	—	**0.5515/0.6139/0.6625**
GS	1.00/1.00/1.00	0.9987/0.9994/0.9995	1.0000/1.0000/1.0000	1.00/1.00/1.00	0.9983/0.9991/0.9995	1.0000/1.0000/1.0000
GS+P	**0.00/0.00/0.00**	**0.5358/0.5764/0.6099**	**0.7773/0.9244/0.9582**	**0.00/0.00/0.06**	**0.5410/0.5849/0.6139**	**0.7969/0.9108/0.9577**
PRC	1.00/1.00/1.00	1.0000/1.0000/1.0000	—	0.94/1.00/1.00	0.8788/0.9895/1.0000	—
PRC+P	**0.00/0.00/0.01**	**0.4952/0.5000/0.5021**	—	**0.00/0.00/0.03**	**0.4980/0.5027/0.5000**	—
T2SMark	1.00/1.00/1.00	0.9999/1.0000/1.0000	1.0000/1.0000/1.0000	1.00/1.00/1.00	0.9998/1.0000/1.0000	1.0000/1.0000/1.0000
T2SMark+P	**0.00/0.00/0.01**	**0.4966/0.5037/0.5119**	**0.5335/0.6868/0.7626**	**0.00/0.00/0.01**	**0.4921/0.5055/0.5116**	**0.5783/0.6964/0.7551**

5 Experiments

5.1 Experimental Setting

In our main experiments, we employ *Stable Diffusion v2.1* (SD 2.1)[2] as our service provider model. All images are generated at a resolution of 512×512 using classifier-free guidance with a scale of 7.5 and 50 inference steps. All images are generated with prompts from the Stable Diffusion Prompts[3] (SDP) dataset. All of our experiments are run on an NVIDIA L40 GPU.

Watermarking schemes. We evaluate PGID's effectiveness with four semantic watermarking schemes: Tree Ring [4], Gaussian Shading [5], PRC [20], and T2SMark [31]. For each scheme, we compare the performance of its default inversion process against its PGID-augmented variant (denoted "+P"). All schemes that allow embedded bit messages (Gaussian Shading, PRC, and T2SMark) are encoded with 256 bits. Unless specified otherwise, we follow the default settings of the watermarking schemes for embedding and detection. Detailed settings for each method are in Appendix A.1.

Hyperparameter settings. For PGID-R (against removal), we set the stopping timestep $k = 10$, number of skip inversion steps $s = 1$, and guidance strength $\gamma = 0.045$ for all watermarking schemes. For PGID-F (against forgery), we employ $k = 15, s = 3, \gamma = 0.001$ for Gaussian Shading, Tree Ring, and T2SMark. The setting for PRC is the same, with the only difference being $k = 7$. We set T in stages I and III to the default 50. We provide an analysis of different hyperparameter settings at Appendix B.

Imprint attacks. Following Müller et al. (2025) [8], we consider the imprint attacks with 50, 100 and 150 optimization steps. In our main experiments, we use two models as the proxy attacker model: *Stable Diffusion v2.1* (SD 2.1) and *Stable Diffusion v1.5* (SD 1.5), representing the cases where the attacker's model matches and does not match the service provider model, respectively. By default, the DDIM scheduler is used as the attacker's sampler with 50 inversion steps. For the imprint forgery attack, we report results on two different datasets as cover images: ImageNet [38]

[2]https://huggingface.co/Manojb/stable-diffusion-2-1-base
[3]https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts

Table 3: Results against imprint forgery attack on ImageNet dataset for four semantic watermarking schemes and their PGID-augmented variants. Results are reported for the attack at 50/100/150 optimization steps. Lower AUC and Det. indicate higher robustness to the attack. The best result for each scheme is in **bold**.

Method	Attacker Model: SD 2.1			Attacker Model: SD 1.5		
	Det.↓	Bit Acc.↓	AUC↓	Det.↓	Bit Acc.↓	AUC↓
TR	0.99/1.00/1.00	—	0.9994/0.9998/0.9998	0.99/1.00/1.00	—	0.9990/1.0000/1.0000
TR+P	**0.06/0.08/0.12**	—	**0.5642/0.6324/0.6915**	**0.05/0.09/0.11**	—	**0.5653/0.6295/0.7030**
GS	1.00/1.00/1.00	0.9984/0.9992/0.9996	1.0000/1.0000/1.0000	1.00/1.00/1.00	0.9972/0.9986/0.9991	1.0000/1.0000/1.0000
GS+P	**0.00/0.04/0.08**	**0.5479/0.5895/0.6210**	**0.7801/0.8968/0.9288**	**0.01/0.06/0.14**	**0.5512/0.5970/0.6273**	**0.8044/0.8971/0.9367**
PRC	0.99/0.99/1.00	0.9846/0.9951/0.9948	—	0.89/0.98/0.99	0.8815/0.9655/0.9744	—
PRC+P	**0.00/0.00/0.04**	**0.5012/0.4967/0.5048**	—	**0.00/0.00/0.00**	**0.4944/0.5038/0.4986**	—
T2SMark	1.00/1.00/1.00	0.9999/1.0000/1.0000	1.0000/1.0000/1.0000	1.00/1.00/1.00	0.9999/1.0000/1.0000	1.0000/1.0000/1.0000
T2SMark+P	**0.00/0.02/0.06**	**0.5000/0.5143/0.5250**	**0.5339/0.6426/0.7452**	**0.00/0.04/0.07**	**0.5027/0.5114/0.5423**	**0.5593/0.6295/0.7030**

Table 4: Robustness and separability of authentic watermarks after passing through the PGID-F defense. AUC is calculated for the attacks at 50, 100, and 150 steps (50/100/150). Higher Detection Rate, Bit Accuracy, and AUC indicate that the defense preserves the authentic watermark while effectively separating it from forged images.

Method	Det.↑	Bit Acc.↑	AUC↑(MS-COCO)		AUC↑(ImageNet)	
			Atk Model: SD 2.1	Atk Model: SD 1.5	Atk Model: SD 2.1	Atk Model: SD 1.5
TR+P	0.96	—	0.9930/0.9902/0.9851	0.9930/0.9886/0.9854	0.9913/0.9874/0.9805	0.9916/0.9883/0.9802
GS+P	1.00	0.8947	1.0000/1.0000/1.0000	1.0000/1.0000/1.0000	1.0000/0.9996/0.9986	1.0000/0.9992/0.9972
PRC+P	0.93	0.8452	—	—	—	—
T2SMark+P	1.00	0.9591	1.0000/1.0000/1.0000	1.0000/1.0000/0.9998	1.0000/0.9996/0.9970	1.0000/0.9983/0.9940

and MS-COCO [39]. In our main results, for each watermarking method, our evaluation is based on 100 watermarked images, 100 generated unwatermarked images, 100 removal-attacked images, and 100 forged images for each proxy model and cover image dataset.

Evaluation. We benchmark detection effectiveness using AUC and TPR at fixed FPR thresholds (denoted as Det.). To maintain consistency with the original literature, we set the FPR of 10^{-6} for Gaussian Shading, PRC, and T2SMark, while Tree Ring is evaluated at the FPR of 1%. In addition, we also report the bit accuracy, defined as the ratio of correctly reconstructed bits to the total payload length, for watermarking schemes that support multi-bit message embedding. Notably, since PRC utilizes image-dependent dynamic thresholds rather than a global one, its AUC calculation is omitted. In the removal case, the detection metrics are calculated between generated unwatermarked images and removal-attacked images, meaning higher AUC and TPR indicate higher robustness. Meanwhile, in the forgery case, the detection metrics are calculated between generated unwatermarked images and forged images. Higher AUC and TPR indicate more successful forgery attacks where the forged watermarks are highly distinguishable from the unwatermarked images. We provide more details of the thresholds and evaluations in Appendix A.1.

5.2 Main Results

Against removal attack. Table 1 reports the watermark detection results in the removal case. Baseline schemes using standard inversion are highly vulnerable to the attack, leading to nearly none being able to detect the watermark after removal in both proxy model settings. T2SMark is the only watermarking scheme that remains fairly robust but loses all payload information (~ 0.5 bit accuracy). PGID-R effectively restores these obscured watermarks, achieving nearly 100% detection and AUC across all schemes and attacker models. Furthermore, PGID-R also increases the bit accuracy to nearly perfect bit message retrieval, with the most substantial increase in Gaussian Shading: from roughly 0.00 to nearly 1.00. The AUC for all methods is also near 100%, showing that with PGID-R, originally watermarked images after removal attack can easily be distinguished from unwatermarked images.

Against forgery attack. Table 2 and Table 3 show the results against forgery attacks. All baseline watermarking methods are extremely susceptible to the forgery attack, with nearly 100% of forged images misidentified as watermarked. PGID-F suppresses the forged signal, reducing detection rates

Table 5: Results against imprint removal attack using *PixArt-*α as the service provider model.

Method	Attacker Model: SD 2.1		
	Det.↑	Bit Acc.↑	AUC↑
TR	0.20/0.06/0.03	—	0.4170/0.2103/0.1307
TR+P	**0.93/1.00/1.00**	—	**0.9867/1.0000/0.9978**
GS	0.03/0.00/0.00	0.4444/0.0927/0.0264	0.3608/0.0000/0.0000
GS+P	**1.00/1.00/1.00**	**0.9178/0.9846/0.9932**	**1.0000/1.0000/1.0000**
PRC	0.50/0.00/0.00	0.5439/0.5033/0.5013	—
PRC+P	**1.00/0.97/1.00**	**1.0000/0.7085/0.9475**	—
T2SMark	0.30/0.97/1.00	0.5434/0.4932/0.4956	0.9020/**1.0000/1.0000**
T2SMark+P	**0.93/1.00/1.00**	**0.9298/0.9957/0.9975**	**0.9933/1.0000/1.0000**

Table 6: Results against imprint forgery attack using *PixArt-*α as the service provider model.

Method	Attacker Model: SD 2.1		
	Det.↓	Bit Acc.↓	AUC↓
TR	0.97/0.97/0.97	—	0.9967/0.9993/0.9993
TR+P	**0.10/0.17/0.23**	—	**0.5178/0.5811/0.6367**
GS	1.00/1.00/1.00	0.9772/0.9893/0.9943	1.0000/1.0000/1.0000
GS+P	**0.00/0.07/0.27**	**0.5715/0.6240/0.6655**	**0.9322/0.9772/0.9872**
PRC	0.97/1.00/1.00	0.9858/1.0000/1.0000	—
PRC+P	**0.00/0.00/0.00**	**0.5055/0.4960/0.4964**	—
T2SMark	1.00/1.00/1.00	0.9905/0.9967/0.9979	1.0000/1.0000/1.0000
T2SMark+P	**0.00/0.00/0.13**	**0.5016/0.5435/0.6301**	**0.0056/0.2378/0.3833**

to near zero and bit accuracy to random-guess levels (~ 0.5). Generally, the AUC for all schemes also drops when using PGID-F. This shows that with PGID-F, the test statistics of forged images are re-aligned to be closer to those of unwatermarked images. In the case of Gaussian Shading, while the forged and unwatermarked distributions remain mathematically separable (reflected in a high AUC), most forged images are no longer detected as watermarked. This shows that PGID-F successfully degrades the forged imprint below the detection threshold, effectively securing the system.

5.3 Authentic Watermark Robustness and Separability

We further analyze our method's ability to separate authentic watermarks from forged ones by running watermarked images through PGID-F, as shown in Table 4. We report the same metrics as the main experiment, with the only difference being the AUC. The AUC in Table 4 is calculated between watermarked images and forged images from MS-COCO and ImageNet. As shown, PGID-F consistently preserves the authentic watermarks, maintaining detection rates above 0.93 and high bit accuracies across all schemes. Furthermore, the defense achieves near-perfect AUC scores against both SD 2.1 and SD 1.5 attacker models across all optimization steps. This demonstrates that our method can reliably isolate genuine watermarks from forged ones.

5.4 Generalizability on Other Architectures

To test the application of our method to other architectures, we utilize *PixArt-*α[4] [40], a model with a diffusion transformer backbone as our service provider model, and SD 2.1 as the attacker model. We report the same metrics as in the main experiments. The metrics are evaluated based on 30 generated unwatermarked images, 30 watermarked, and 30 attacked images from each strategy. Cover images are taken from the MS-COCO dataset. We use a different hyperparameter setting from our main experiments. Details about our settings are provided in Appendix A.3.

Table 7: Robustness and separability of authentic watermarks after passing through the PGID-F defense, using *PixArt-*α as the service provider model.

Method	Det.↑	Bit Acc.↑	Attacker: SD 2.1
			AUC↑
TR+P	0.93	—	0.9822/0.9722/0.9644
GS+P	1.00	0.9313	1.0000/1.0000/0.9978
PRC+P	1.00	0.6337	—
T2SMark+P	0.93	0.9780	1.0000/0.9922/0.9711

Results are presented in Table 5, Table 6 and Table 7. Overall, PGID retains similar performance when applied to a DiT-based model. Against removal, PGID-R continues to restore the watermark signals with high accuracy, exhibiting only a modest performance degradation compared to the main results. In the forgery case, although there is a decrease in protection against attacks with higher optimization steps, PGID-F retains its capability to differentiate between forged and authentic watermarked instances, reflected by the high AUC depicted in Table 7. These results highlight the plug-and-play and training-free features of our method.

6 Conclusion

In this paper, we propose a novel plug-and-play, training-free unified framework capable of protecting semantic watermark methods against both imprint removal and forgery attacks. Based on the observation that imprint attacks maliciously displace the latents out of their respective watermarked

[4]https://huggingface.co/PixArt-alpha/PixArt-XL-2-512x512

or unwatermarked region, our method effectively projects the attacked latents back to their original region. Our experiments demonstrate that PGID effectively restores obscured watermarks against removal attacks and reliably isolates forged images from authentic ones across multiple state-of-the-art watermarking schemes and different model architectures. However, we acknowledge that applying our method to different schemes and architectures might require users to tailor hyperparameters to their specific setup rather than relying on a single default to fully optimize the defense. Despite that, our novel approach opens the path for more robust and reliable content provenance against adversarial threats in watermarking systems.

References

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

[2] 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, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[3] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 12606–12633. PMLR / OpenReview.net, 2024.

[4] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Invisible fingerprints for diffusion images. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[5] Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[6] Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XXVIII*, Lecture Notes in Computer Science, pages 338–354. Springer, 2024.

[7] Kasra Arabi, R Teal Witter, Chinmay Hegde, and Niv Cohen. Seal: Semantic aware image watermarking. *arXiv preprint arXiv:2503.12172*, 2025.

[8] Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 20937–20946, June 2025.

[9] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[10] Yuwei Chen, Zhenliang He, Jia Tang, Meina Kan, and Shiguang Shan. One-step inversion excels in extracting diffusion watermarks. *arXiv preprint arXiv:2602.09494*, 2026.

[11] Hanyi Wang, Han Fang, Shi-Lin Wang, and Ee-Chien Chang. Roar: Reducing inversion error in generative image watermarking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[12] Ingemar J Cox, Matthew L Miller, Jeffrey A Bloom, Jessica Fridrich, and Ton Kalker. Digital watermarking. *Morgan Kaufmann Publishers*, 54(56-59):2, 2008.

[13] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019.

[14] Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM international conference on multimedia*, pages 41–49, 2021.

[15] Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1532–1542, 2022.

[16] Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Can simple averaging defeat modern watermarks? 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.

[17] Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. Benchmarking the robustness of image watermarks. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 1456–1492, 2024.

[18] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. *ICCV*, 2023.

[19] Rui Min, Sen Li, Hongyang Chen, and Minhao Cheng. A watermark-conditioned diffusion model for ip protection. In *European Conference on Computer Vision*, pages 104–120. Springer, 2024.

[20] 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.

[21] Frank Hartung and Martin Kutter. Multimedia watermarking techniques. *Proceedings of the IEEE*, 87(7):1079–1107, 1999.

[22] S. Voloshynovskiy, S. Pereira, T. Pun, J.J. Eggers, and J.K. Su. Attacks on digital watermarks: classification, estimation based attacks, and benchmarks. *IEEE Communications Magazine*, 39(8):118–126, 2001.

[23] Yuepeng Hu, Zhengyuan Jiang, Moyang Guo, and Neil Zhenqiang Gong. A transfer attack to image watermarks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

[24] 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*, 2024.

[25] Yepeng Liu, Yiren Song, Hai Ci, Yu Zhang, Haofan Wang, Mike Zheng Shou, and Yuheng Bu. Image watermarks are removable using controllable regeneration from clean noise. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

[26] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

[27] Chun Tong Lei, Hon Ming Yam, Zhongliang Guo, Yifei Qian, and Chun Pong Lau. Instant adversarial purification with adversarial consistency distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24331–24340, 2025.

[28] Ziping Dong, Chao Shuai, Zhongjie Ba, Peng Cheng, Zhan Qin, Qinglong Wang, and Kui Ren. Wmcopier: Forging invisible image watermarks on arbitrary images. *arXiv preprint arXiv:2503.22330*, 2025.

[29] Tomás Soucek, Sylvestre-Alvise Rebuffi, Pierre Fernandez, Nikola Jovanovic, Hady Elsahar, Valeriu Lacatusu, Tuan Tran, and Alexandre Mourachko. Transferable black-box one-shot forging of watermarks via image preference models. *CoRR*, abs/2510.20468, 2025.

[30] Kasra Arabi, Benjamin Feuer, R. Teal Witter, Chinmay Hegde, and Niv Cohen. Hidden in the noise: Two-stage robust watermarking for images. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

[31] Jindong Yang, Han Fang, Weiming Zhang, Nenghai Yu, and Kejiang Chen. T2smark: Balancing robustness and diversity in noise-as-watermark for diffusion models. *arXiv preprint arXiv:2510.22366*, 2025.

[32] Xin Zhang, Zijin Yang, Kejiang Chen, Linfeng Ma, Weiming Zhang, and Nenghai Yu. Sembind: Binding diffusion watermarks to semantics against black-box forgery attacks. *arXiv preprint arXiv:2601.20310*, 2026.

[33] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[34] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

[35] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[36] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[37] Anubhav Jain, Yuya Kobayashi, Naoki Murata, Yuhta Takida, Takashi Shibuya, Yuki Mitsufuji, Niv Cohen, Nasir D. Memon, and Julian Togelius. Forging and removing latent-noise diffusion watermarks using a single image. *CoRR*, abs/2504.20111, 2025.

[38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[40] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James T. Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[41] Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. *arXiv preprint arXiv:2402.09370*, 2024.

[42] Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hyewon Bae, and Se Young Chun. On exact inversion of dpm-solvers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 7069–7078. IEEE, 2024.

[43] Huming Qiu, Zhaoxiang Wang, Mi Zhang, Xiaohan Zhang, Xiaoyu You, and Min Yang. The future unmarked: Watermark removal in AI-generated images via next-frame prediction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Table of Contents: Appendix

A Implementation Details **14**

A.1 Semantic Watermarking 14

A.2 Imprint Attacks 15

A.3 Hyperparameter settings: 16

B Ablation Study of Hyperparameter Settings **16**

B.1 Number of skip inversions s 16

B.2 Stopping Timestep k 18

B.3 Guidance Strength γ 19

B.4 Summary 19

C Computational Analysis **20**

D Against Different Attack Scheduler **21**

E Transferability to Other Attacks **21**

F Evaluation on Larger Test Sets **23**

G Limitations **24**

A Implementation Details

A.1 Semantic Watermarking

In this section, we present more details on the evaluated semantic watermarking methods and our implementation settings. Figure 5 describes the main methodology of semantic watermarks and imprint attacks.

Tree Ring. Tree-Ring watermarking [4] is the pioneering work of the semantic watermark paradigm. It embeds a structured pattern into the Fourier space of the initial noise vector used during the diffusion sampling process, allowing the watermark to be robustly detected by inverting the generated image back to its initial latent noise state. After obtaining the estimated initial noise, the method calculates a test statistic by summing the squared absolute differences between the observed and expected frequency values within the embedded rings. Under the null hypothesis (H_0) that an unwatermarked image's inverted initial noise follows a Gaussian distribution with an unknown variance, the test statistic conforms to a noncentral χ^2 distribution, enabling the watermarker to calculate a p-value. If it falls below a predefined significance threshold τ, H_0 is rejected, determining the image as watermarked.

In our experiments, we adhere to the original paper by embedding a ring pattern using a circular mask with a radius of 10 and employing DDIM as the scheduler. As the threshold is not reported in the original paper, for our experiments, we compute the threshold τ from 5,000 generated unwatermarked images for the desired FPR. The thresholds are set at 0.05385 and 0.04743, corresponding to SD 2.1 and PixArt-α for FPR = 1%, respectively. For AUC calculation, we use the p-value as the continuous score.

Gaussian Shading. Unlike Tree-Ring, which causes the initial noise to deviate from a Gaussian distribution, resulting in performance loss, Gaussian Shading [5] embeds watermark information while preserving the noise distribution. Here we focus on the standard setting of $l = 1$ from the original paper. In this setting, an encrypted message m derived from the original bit message s determines the signs of the initial noise. The watermark is detected by inverting the image, decrypting the retrieved message s', and comparing it to reference messages.

Under the null hypothesis of an unwatermarked image, the extracted bits follow a Bernoulli distribution with parameter $p = 0.5$. When comparing s' to a single reference message of length k, the integer count of matching bits, denoted as c, follows a binomial distribution $B(k, 0.5)$. The probability of observing more than c_τ matching bits by chance is given by the regularized incomplete beta function:

$$\text{FPR}(c_\tau) = P(c > c_\tau) = B_{1/2}(c_\tau + 1, k - c_\tau) \tag{4}$$

In zero-bit watermarking, s' is compared to a single predefined message, bounded directly by $\text{FPR}(c_\tau)$. However, in multi-bit watermarking, s' is compared against N possible user IDs known to the service provider, and the match with the highest accuracy is selected. Because this constitutes a multiple testing problem, the overall False Positive Rate is adjusted depending on the number of users N:

$$\text{FPR}(c_\tau, N) = 1 - (1 - \text{FPR}(c_\tau))^N \tag{5}$$

For evaluation, we follow Müller et al. (2025) [8] and map the integer count threshold c_τ to a bit accuracy threshold, $\tau = \frac{c_\tau}{k}$. In our experiments, we focus on the multi-watermarking scenario with $N = 100,000$ as in [5] and target $\text{FPR}(c_\tau, N) = 10^{-6}$, which yields $\tau = 0.70703$ [8]. We use the DDIM sampler and parameter settings from Yang et al. (2024)[5] with $k = 256$, $\rho = 64$, and $l = 1$. For AUC calculation, we use the bit accuracy as the continuous score.

PRC. The PRC watermark builds upon Gaussian Shading by introducing a pseudorandom error-correcting code (PRC) [41] for better undetectability. This method embeds a message by sampling the initial noise vector directly from a cryptographically secure codebook, ensuring the watermarked latents are computationally indistinguishable from standard Gaussian noise. Detection follows the standard inversion paradigm to recover the estimated initial noise. However, unlike previous methods that use a globally fixed threshold, the PRC watermark utilizes an image-dependent detection logic based on soft-decision decoding. Specifically, the decoding process assigns confidence weights to the estimated initial noise values based on their magnitudes, treating larger values as more reliable indicators of the watermark. Based on these weights, a dynamic threshold τ is calculated for each image using Hoeffding's inequality to strictly bound the False Positive Rate (FPR).

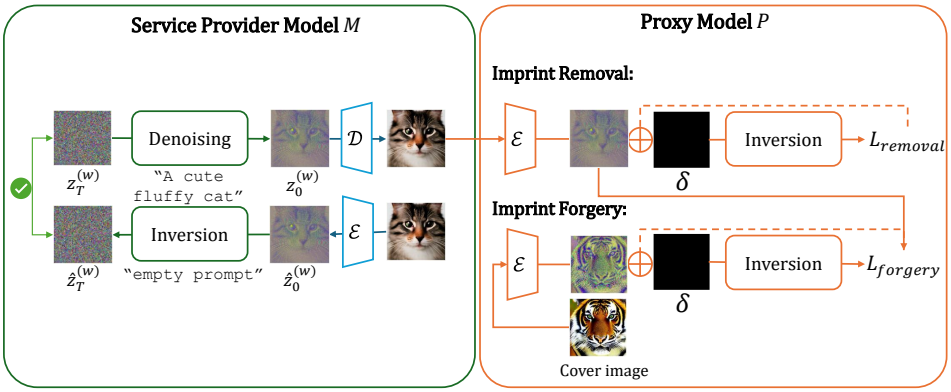


Figure 5: The left side presents the semantic watermarking paradigm. The initial noise contains a watermark pattern, which can be verified after inversion. The right side illustrates imprint removal and forgery attacks. From a proxy diffusion model P, the attacks optimize a perturbation and add it to either a watermarked image or a cover image in latent space to yield deceptive results.

Following the settings in Gunn et al. (2025)[20], we implement PRC using the DPM scheduler for generation and the exact inversion method from Hong et al. (2023)[42] for inversion. Because the dynamically calculated threshold varies across images, there is no globally comparable continuous test statistic. Consequently, we omit the AUC for the PRC scheme and instead evaluate its performance using the True Positive Rate (TPR) at a targeted FPR of 10^{-6}.

T2SMark. T2SMark focuses on resolving the fundamental trade-off between watermark robustness and generation diversity. To achieve this, it introduces Tail-Truncated Sampling (TTS), which selectively embeds watermark information exclusively into the more reliable tail regions of the Gaussian distribution. The central region is left for random sampling to maintain the overall noise distribution and preserve sample diversity. Furthermore, it employs a two-stage hierarchical key encryption framework to randomize the codewords. For detection, following the standard inversion process, the estimated initial noise vector is projected onto the normal vectors of the first stage. The detection test statistic is computed as the L_1 norm of this first-stage projection vector. Under the null hypothesis of an unwatermarked image, this statistic is compared against a threshold d that is calibrated to a target False Positive Rate (FPR). If the test statistic exceeds d, the image is determined to be watermarked.

In our experiments, we follow the default settings from the original paper, including the use of a DDIM scheduler and the empirically optimal truncation threshold $\tau = 0.674$. For detection evaluation, the threshold d is calibrated to achieve a target FPR of 10^{-6} to remain consistent with the original work. Since the original paper does not report the threshold d, we empirically estimate the null distribution using 10,000 unwatermarked images. By assuming normality and computing the critical value at the 10^{-6} upper tail, we obtain $d = 260.7018$ and $d = 250.6571$ for SD 2.1 and PixArt-α, respectively. We use the test statistic, which is the L_1 norm of the first-stage projection vector, for the calculation of the AUC.

A.2 Imprint Attacks

Imprint Removal Attack. Proposed by Müller et al. (2025)[8], the attack aims to remove the watermark from a watermarked image $x^{(w)}$ by using a proxy model and optimizing a perturbation δ in the latent space with the following loss function:

$$\mathcal{L}_{\text{removal}} = \|I_{0\to T}(\hat{z}_0^{(w)} + \delta) + \hat{z}_T^{(w)}\|_2 \tag{6}$$

where $I_{0\to T}(\cdot)$ is the diffusion inversion process using the proxy model, $\hat{z}_T^{(w)}$ is the estimated initial noise from the watermarked image, and $\hat{z}_0^{(w)} = \mathcal{E}(x^{(w)})$ with $\mathcal{E}$ being the encoder from the proxy model. The attack pushes the estimated initial noise away from its original value. The final image with the watermark removed is obtained by decoding the perturbed latent: $x_{removed}^{(w)} = \mathcal{D}(\hat{z}_0^{(w)} + \delta)$.

Imprint Forgery Attack. On the other hand, the imprint forgery attack aims to imprint the watermark from a watermarked image $x^{(w)}$ to a cover image $x^{(c)}$ by optimizing a perturbation δ in the latent space with the following loss function:

$$\mathcal{L}_{\text{forgery}} = \|I_{0\to T}(\hat{z}_0^{(c)} + \delta) - \hat{z}_T^{(w)}\|_2 \tag{7}$$

where $I_{0\to T}(\cdot)$ is the diffusion inversion process using the proxy model, $\hat{z}_T^{(w)}$ is the estimated initial noise from the watermarked image, and $\hat{z}_0^{(c)} = \mathcal{E}(x^{(c)})$ with $\mathcal{E}$ being the encoder from the proxy model. Opposite to the removal attack, the introduced perturbation guides the estimated initial noise of the cover image towards the watermarked image, thereby fooling the watermark detector. The final forged image is obtained by decoding the perturbed latent: $x_{forged}^{(c)} = \mathcal{D}(\hat{z}_0^{(c)} + \delta)$.

Following Müller et al. (2025)[8], we report the results on 50, 100, and 150 optimization steps with a learning rate of 0.01. In our main experiments, we utilize DDIM as the attacker's scheduler and consider both white-box and grey-box scenarios where the proxy model used by the attacker is either the same as or different from the service provider model.

A.3 Hyperparameter settings:

SD 2.1 (Main experiments). With SD 2.1 as the service provider model, we follow the settings:

- PGID-R (Against removal): We set the stopping timestep $k = 10$, number of skip inversions $s = 1$, and guidance strength $\gamma = 0.045$ for all watermarking schemes.
- PGID-F (Against forgery): We set the stopping timestep $k = 15$, number of skip inversions $s = 3$, and guidance strength $\gamma = 0.001$ for Tree Ring, Gaussian Shading, and T2SMark. We set the stopping timestep $k = 7$, number of skip inversions $s = 3$, and guidance strength $\gamma = 0.001$ for PRC.

PixArt-α. With PixArt-α as the service provider model, we report the results with the settings:

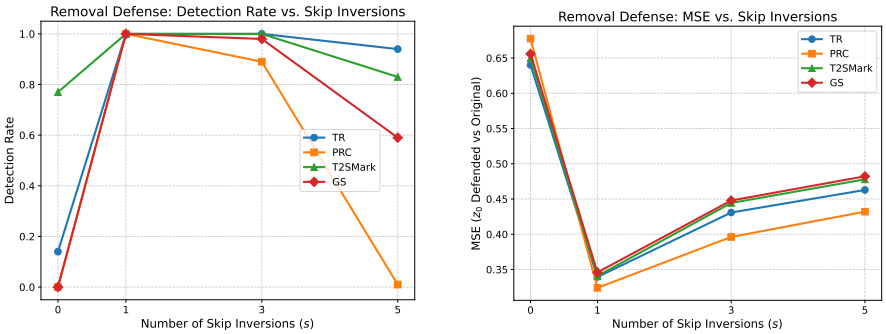
- PGID-R (Against removal): We set the stopping timestep $k = 10$, number of skip inversions $s = 1$, and guidance strength $\gamma = 0.045$ for all watermarking schemes. An exception is that for PRC, we evaluate the attack at 50 optimization steps with the setting: $k = 10$, $s = 1$, and $\gamma = 0.01$. We discuss the rationale for this specialized configuration in Section B.3.
- PGID-F (Against forgery): We set the stopping timestep $k = 8$, number of skip inversions $s = 3$, and guidance strength $\gamma = 0.005$ for all schemes.

B Ablation Study of Hyperparameter Settings

B.1 Number of skip inversions s

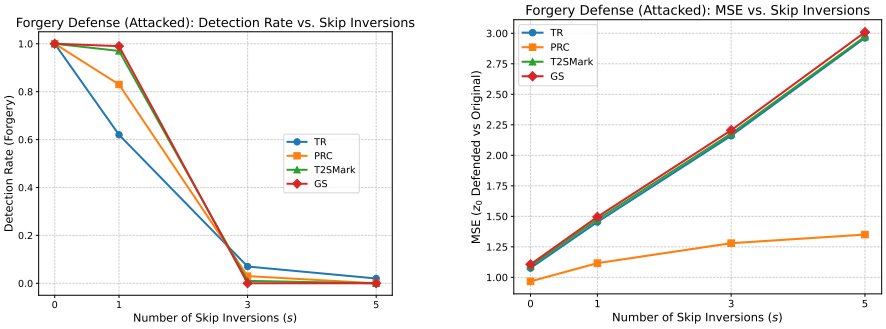
In this section, we justify the necessity of asymmetric inversion-denoising cycles and show the effects of the number of skip inversions s. We evaluate the impact of the skip parameter s using the following experiment configuration: SD 2.1 for both the service provider and the attacker, with imprint attacks optimized over 150 steps. We set the stopping timestep k and guidance strength γ to the values as in our main experiments. Keeping the stopping timestep k and guidance strength γ constant, we vary $s \in \{0, 1, 3, 5\}$ to quantify how cycle asymmetry affects performance.

Against removal attack. As described in Figure 6, symmetric inversion-denoising cycles ($s = 0$) provide nearly no defense for Tree Ring and Gaussian Shading. The setting where $s = 1$ achieves the highest detection rate and also the lowest MSE, confirming the need to denoise more steps than inverting. We attribute this improvement to the perturbed nature of the attacked latent. The introduced perturbation essentially acts as a layer of noise on the original latent. Therefore, extra denoising steps leverage the model's natural ability to remove noise, effectively filtering out the adversarial perturbations and pulling the displaced latent back toward the clean latent manifold. However, with higher s, the restoration capability of PGID could potentially become worse. This occurs because excessive denoising over-smooths the latent and causes the model's natural projection to strip away the embedded watermark signal. From Figure 6a, with $s = 3$, the performance for PRC starts to degrade, while other schemes remain nearly unchanged. Only at $s = 5$, the performance starts to



(a) (b)

Figure 6: (a) Detection rate of removal-attacked images under PGID-R with different numbers of skip inversions s. (b) MSE between the watermarked latent and the projected latent of watermarking schemes under PGID-R with different numbers of skip inversions. The lowest is achieved at $s = 1$, indicating that at this setting, the removal-attacked latent is the closest to the watermarked latent after projection.

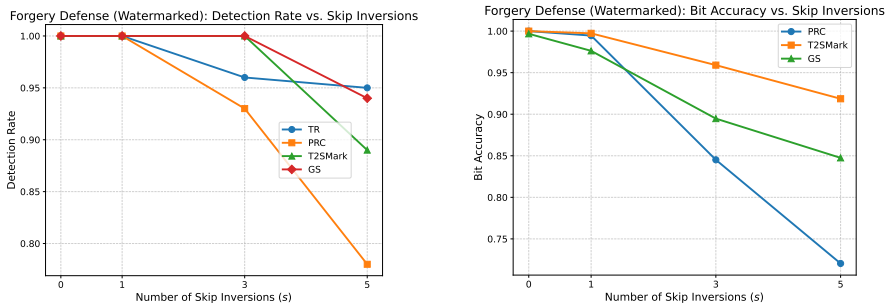


(a) (b)

Figure 7: (a) Detection rate of forged images under PGID-F with different numbers of skip inversions s. (b) MSE between watermarked latent and projected latent under PGID-F with different numbers of skip inversions. Initially, forged latents have a small MSE from the watermarked latent. After projection, a higher MSE indicates that the projected latent is pushed away from the watermarked latent, mitigating the forged signals.

degrade for other schemes. This can be explained by the sensitivity of each watermarking scheme. PRC employs a more intricate detection mechanism than other schemes, which effectively narrows its watermarked region for a given signal. In other words, PRC is more unforgiving, where a slight difference in the projected latent can push it outside of the watermarked region. Generally, no schemes are affected significantly if s stays close to 1.

Against forgery attack. From Figure 7, similar to the removal case, setting $s = 0$ yields no defense for all schemes (the detection rate remains 100%). Our results also show that defense against forgery requires more aggressive latent projection, where defense becomes effective only from $s = 3$ with the current stopping timestep k setting. At $s = 3$, most forged watermarks are rejected, and the high MSE between the watermarked latent and the projected latent reflects this. However, setting s too high comes at the cost of potentially rejecting authentic watermarks as well if the threshold is set at a very low FPR. As shown in Figure 8, even though semantic watermarks are inherently robust to the process, some schemes are less robust than others. From the detection rate and bit accuracy of watermarked images, we found PRC and Tree Ring to be the least robust, while Gaussian Shading and T2SMark are the most robust. This claim is supported in Yang et al. (2025)[31], where they also showed that PRC and Tree Ring can easily lose their watermark from basic image transformations.



(a) (b)

Figure 8: (a) Detection rate of watermarked images under PGID-F with different numbers of skip inversions s. (b) Bit accuracy of watermarked images under PGID-F with different numbers of skip inversions. With a high value of s, watermarks from schemes that are not very robust (Tree Ring, PRC) could be falsely denied.

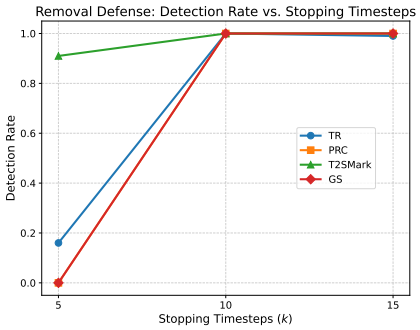


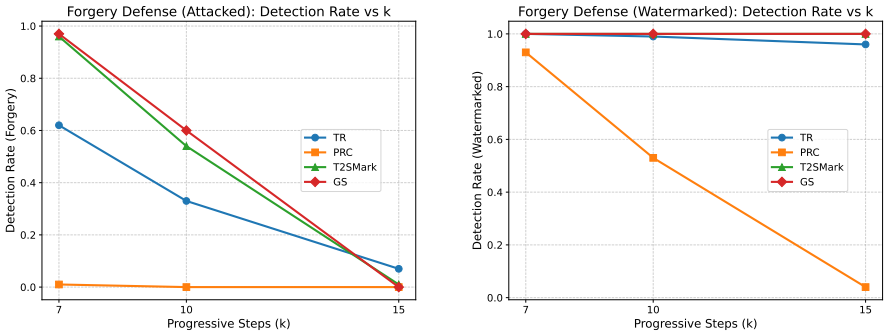
Figure 9: Detection rate of removal-attacked images under PGID-R with different stopping timesteps k. With k too small, the latent projection fails to provide good results. However, with sufficiently large k, nearly perfect detection rate is achieved.

B.2 Stopping Timestep k

With the same experiment configuration, in this part, we analyse the effects of the stopping timestep k by fixing the number of inversions s and guidance strength γ. We vary $k \in \{5, 10, 15\}$ for removal case and $k \in \{7, 10, 15\}$ in forgery case.

Against removal attack. Figure 9 illustrates the detection rate of images subjected to removal attack under different stopping timesteps k. At $k = 5$, most schemes remain unable to detect watermarks. This is because when k is too small, the latent refinement process is confined to the initial timesteps. Consequently, it lacks the necessary leverage to correct the overall inversion trajectory through to the final timestep T. Figure 9 indicates that a near-perfect detection rate is achieved by both $k = 10$ and $k = 15$. This shows that PGID-R can restore removed watermark signals with a sufficiently large k.

Against forgery attack. From Figure 10a, it can be observed that with k large enough, most of the forged watermark instances are no longer detectable. This can easily be explained similarly to the removal case: with higher k, the latent refinement process has more leverage to correct the inversion trajectory. Most forged instances from Gaussian Shading, Tree Ring, and T2SMark are projected back to the unwatermarked region at $k = 15$. PRC is a special case where forged images lose all of their signal with only $k = 7$. As described in Figure 10b, other schemes maintain a high detection rate at higher k, while for PRC, the detection rate degrades heavily. Again, we attribute this to the inherent low robustness of PRC, which makes even forged watermarks easy to remove. As a result, a user should consider a suitable choice of k depending on the robustness of the scheme they are using. Setting k can be thought of as a trade-off: a higher k facilitates rejection of forged instances, but could also potentially reject some authentic watermarked instances at a low FPR threshold setting.



(a) (b)

Figure 10: (a) Detection rate of forged images under PGID-F with different stopping timesteps k. (b) Detection rate of watermarked images under PGID-F with different stopping timesteps. With a higher k, more forged instances are rejected. However, with a low FPR threshold, authentic watermarks could also be rejected if k is too high. Therefore, k should be selected based on this trade-off.

B.3 Guidance Strength γ

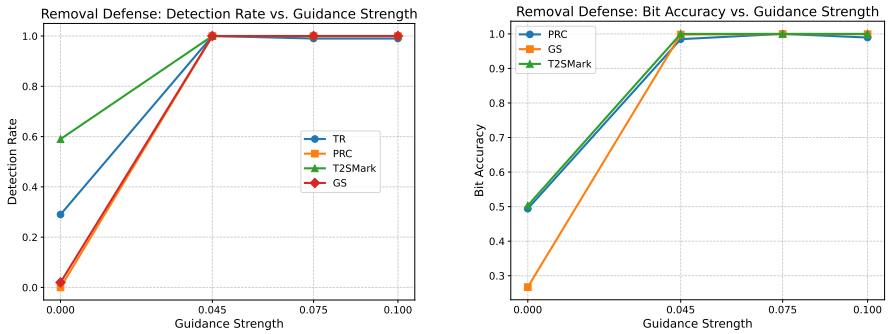
To better understand the role of the guidance strength γ in our refinement process, with the same experiment configuration, we fix the stopping timestep k and the number of inversions s while varying the value of γ. We consider the values $\gamma \in \{0, 0.045, 0.075, 0.1\}$ in removal defense, and $\gamma \in \{0, 0.001, 0.005, 0.01\}$ in forgery defense. $\gamma = 0$ is equivalent to the scenario where no guidance is introduced.

Against removal attack. Figure 11 indicates that the introduced guidance provides superior performance compared to the configuration without guidance. The choice of γ could also be relatively versatile, where a higher γ yields similar results to the setting in the main experiments ($\gamma = 0.045$). However, for schemes with very high sensitivity, such as PRC, we witness a trade-off between the protection against the attack at lower and higher optimization steps. As described in Figure 12, with smaller values of γ ($\gamma = 0.025$), it might defend against the attack at 50 steps, but is unable to retrieve the watermarks against an attack with higher steps. On the other hand, with larger values of γ ($\gamma = 0.075$), only attacks with larger optimization steps can be defended. This is a phenomenon we only observe with PRC. We believe that the high sensitivity of PRC demands a much more precise latent projection compared to other schemes. Consequently, securing such sensitive schemes against varying attack intensities either requires extensive hyperparameter tuning to find a globally optimal γ, or leveraging multiple parallel PGID instances configured with diverse settings to ensure comprehensive protection. In our experiments with PixArt-α, we follow the second approach.

Against forgery attack. In contrast to the removal case, we observe that the guidance strength γ plays a less critical role in defending against forgery. This is because the latent projection of forged images is predominantly driven by the diffusion model's intrinsic manifold projection capabilities. Nevertheless, γ provides an additional dimension of control to fine-tune the separation between authentic and forged distributions. Generally, as shown from Tree Ring and PRC in Figure 13, higher guidance strength facilitates forged instances to lose watermarks. However, with high guidance strength, some schemes can have erratic behaviors. From Figure 13, the detection rate of Gaussian Shading and T2SMark fluctuates as the guidance strength increases. Furthermore, an excessively high guidance strength also risks rejecting authentic watermarks, as shown in Figure 14.

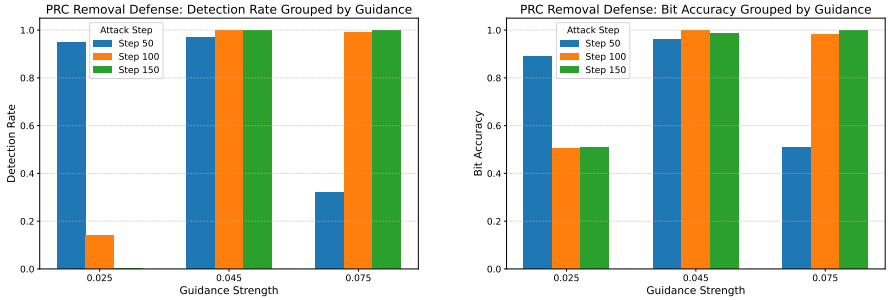
B.4 Summary

Overall, in this section, we first demonstrated the critical effect of skip inversions on the performance of PGID. Then we showed that with a k large enough and suitable s and γ, one can restore removed watermarks consistently. We also discussed the need for a high k for forgery, and the trade-off based on the robustness of the watermarking scheme used. Next, we discussed the effects of guidance strength, where it is crucial in removal, but not as important in forgery. We want to briefly note that since watermarking schemes behave differently when applied to different model architectures, the working hyperparameter settings of one model might not transfer to another. Finally, we want to mention



(a) (b)

Figure 11: (a) Detection rate of removal-attacked images under PGID-R with different guidance strength γ. (b) Bit accuracy of removal-attacked images under PGID-R with different guidance strength γ. With guidance, the performance is significantly better, highlighting the need for this mechanism. The guidance strength γ can also be flexible, where multiple values could provide similarly good results.



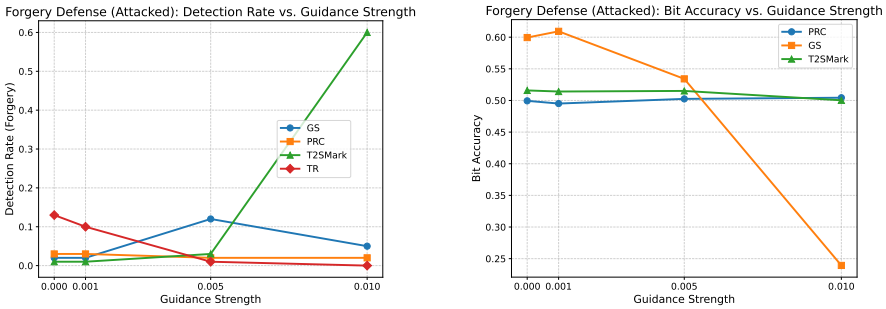
(a) (b)

Figure 12: (a) Detection rate of removal-attacked images at 50, 100, and 150 optimization steps on PRC under PGID-R with different guidance strength γ. (b) Bit accuracy of removal-attacked images at 50, 100, and 150 optimization steps on PRC under PGID-R with different guidance strength γ. In the case of PRC, there is a trade-off between protection against the attack at lower and higher optimization steps, where a lower γ is better against lower steps, and a higher γ is better against higher steps.

that in our current evaluation, the thresholds are taken from a desired FPR of the watermarking scheme. However, as shown from the near-perfect AUC in Table 4, the detection scores for forged and authentic watermarks can be well separated with a good choice of the threshold. Therefore, users could choose a threshold for PGID based on their preferred TPR and FPR to classify authentic and forged watermarks.

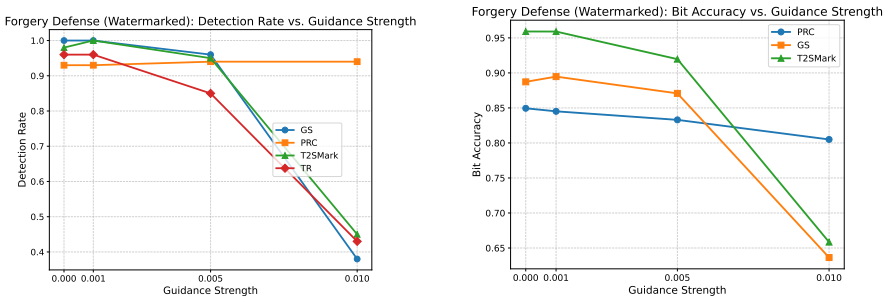
C Computational Analysis

In this section, we report the runtime comparison between our method and the standard inversion process using DDIM. Our results, as shown in Table 8, indicate that we introduce computational overhead compared to the standard one-stage DDIM inversion. This overhead primarily arises from the progressive latent refinement in the second stage, where a larger k and smaller s result in increased runtime. However, we argue that this increase is a necessary and acceptable trade-off for system security: while standard inversion is faster, it completely fails against the introduced threats. Furthermore, the computational overhead of PGID is highly feasible and manageable, as service providers only need to apply the framework to suspicious images.



(a) (b)

Figure 13: (a) Detection rate of forged images under PGID-F with different guidance strength γ. (b) Bit accuracy of forged images under PGID-F with different guidance strength γ. There is a marginal difference between the performance without guidance and at $\gamma = 0.001$, with the setting $\gamma = 0.001$ being slightly better. At higher guidance strengths, the performance became erratic.



(a) (b)

Figure 14: (a) Detection rate of watermarked images under PGID-F with different guidance strength γ. (b) Bit accuracy of watermarked images under PGID-F with different guidance strength γ. Guidance strength γ being too high leads to watermarked images losing their watermarks.

D Against Different Attack Scheduler

In our primary evaluation, we followed the standard setting of using the DDIM scheduler for the attacker's proxy model. In this section, we consider the case where the attacker utilizes DPM as the scheduler. We report the detection metrics similar to the main experiments. SD 2.1 is employed as both the service provider model and the attacker's proxy model. For cover images in the forgery case, we use the same set of images from MS-COCO as in the main experiments. Our hyperparameter settings remain the same as in our main experiments.

As shown in Table 9, Table 10, and Table 11, we achieve similar results to when the attacker uses DDIM. PGID-R is still capable of restoring the removed watermarks at an almost perfect detection rate, and PGID-F retains its forgery detection effectiveness by suppressing forged signals to random-guess levels. We conclude that the attacker's scheduler choice does not affect PGID's performance.

E Transferability to Other Attacks

Even though PGID is designed to be effective against imprint forgery and removal attacks, it could also provide defense against other perturbation-based attacks. In this section, we consider two attacks proposed by Jain et al. (2025)[37] and Yang et al. (2024)[16], which we refer to as the VAE attack and the Averaging attack, respectively. However, it is notable that these two attacks are not as effective and consistent as imprint attacks. From our experiments, we found that the VAE attack only works in the forgery case on three out of the four schemes (Tree Ring, Gaussian Shading, and T2SMark), while the Averaging attack only works on Tree Ring. Previous claims from Yang et al.(2024)[16]

Table 8: Runtime benchmark in milliseconds (ms). We compare standard DDIM inversion to PGID across three stages with the settings in our main results. All of DDIM, Stage I, and Stage III use 50 inference steps.

Pipeline / Stage	**Stage I**	**Stage II**	**Stage III**	**Total (ms)**
DDIM	–	–	–	2039
PGID-R ($k = 10$, $s = 1$)	2039	4086	2040	8165
PGID-F ($k = 15$, $s = 3$)	2040	8092	2041	12172
PGID-F ($k = 7$, $s = 3$)	2039	1553	2039	5631

Table 9: Results against imprint removal attack when the attacker utilizes the DPM scheduler. Results are reported for the attack at 50/100/150 optimization steps. Higher AUC and Det. indicate higher robustness to the attack. The best result for each scheme is in **bold**.

Method	Attacker Scheduler: DPM (SD 2.1)		
	Det.↑	Bit Acc.↑	AUC↑
TR	0.25/0.13/0.10	—	0.4033/0.1877/0.1384
TR+P	**1.00/1.00/1.00**	—	**0.9998/1.0000/1.0000**
GS	0.00/0.00/0.00	0.2586/0.0326/0.0062	0.0734/0.0000/0.0000
GS+P	**1.00/1.00/1.00**	**0.9979/0.9990/0.9990**	**1.0000/1.0000/1.0000**
PRC	0.00/0.00/0.00	0.4976/0.4994/0.4999	—
PRC+P	**0.97/1.00/0.99**	**0.8748/1.0000/0.9753**	—
T2SMark	0.49/0.95/**1.00**	0.5111/0.4945/0.4975	0.8481/0.9993/**1.0000**
T2SMark+P	**1.00/1.00/1.00**	**0.9998/1.0000/1.0000**	**1.0000/1.0000/1.0000**

about their method working on Gaussian Shading has been disproven in Müller et al. (2025) [8], and we observe similar results. Therefore, we report only the successful attack cases. We will present the results against two attacks: VAE-Forgery on three schemes and Average-Removal on Tree Ring.

VAE-Forgery. Jain et al.(2025)[37] proposed forging the watermarks from a single watermarked image $x^{(w)}$ to a cover image $x^{(c)}$ by optimizing a perturbation δ in the pixel space with the following loss function:

$$\mathcal{L}_{\text{forgery}} = \|\mathcal{E}(x^{(c)} + \delta) - \mathcal{E}(x^{(w)})\|_2 + \lambda\|\delta\|_2 \tag{8}$$

where $\mathcal{E}$ is a proxy encoder used by the attacker. The attack aims to push the latent representation of the cover image closer to a watermarked image. We follow the setting from the paper where $\lambda = 5 \times 10^4$ and perform 100 optimization steps. We follow the implementation in the paper where the proxy VAE from SD 1.4 is used.[5]

Average-Removal. Yang et al.(2024)[16] found that for many watermarking schemes, the watermark signal can be extracted by averaging the residual across many watermarked and unwatermarked images. In their method, the watermark signal can be extracted as:

$$\delta = \frac{1}{N}\left(\sum_{i=1}^{N} x_{w,i} - \sum_{i=1}^{N} x_{c,i}\right), \tag{9}$$

where N is the number of averaged images, $\{x_{c,i}\}_{i\in[N]}$ are clean unwatermarked images, and $\{x_{w,i}\}_{i\in[N]}$ are watermarked images. To perform removal, they subtract the extracted watermark pattern from the watermarked images to create the attacked image x^{atk}:

$$x^{atk} = x^{(w)} - \delta \tag{10}$$

For our evaluation, we averaged over 5,000 watermarked images to extract the watermark pattern. We consider the stronger grey-box scenario of the method, where the extracted pattern is calculated from paired watermarked and unwatermarked images.

Setup and Results. We utilize SD 2.1 as the service provider model and use the same set of hyperparameters as in the main experiments. Metrics are evaluated on 100 images of each type, similar to the main experiments.

[5] https://huggingface.co/CompVis/stable-diffusion-v1-4

Table 10: Results against imprint forgery attack on MS-COCO dataset when the attacker utilizes the DPM scheduler. Results are reported for the attack at 50/100/150 optimization steps. Lower AUC and Det. indicate higher robustness to the attack. The best result for each scheme is in **bold**.

Method	Attacker Scheduler: DPM (SD 2.1)		
	Det.↓	Bit Acc.↓	AUC↓
TR	1.00/1.00/1.00	—	1.0000/1.0000/1.0000
TR+P	**0.01/0.07/0.10**	—	**0.5485/0.6111/0.6651**
GS	1.00/1.00/1.00	0.9992/0.9995/0.9996	1.0000/1.0000/1.0000
GS+P	**0.00/0.00/0.02**	**0.5396/0.5781/0.6093**	**0.7842/0.9203/0.9599**
PRC	0.96/1.00/1.00	0.8964/1.0000/1.0000	—
PRC+P	**0.00/0.00/0.03**	**0.4964/0.5041/0.4951**	—
T2SMark	1.00/1.00/1.00	1.0000/1.0000/1.0000	1.0000/1.0000/1.0000
T2SMark+P	**0.00/0.00/0.01**	**0.4982/0.5072/0.5141**	**0.5294/0.6645/0.7657**

Table 11: Separability of authentic watermarks against forgery attack using the DPM scheduler. AUC is calculated between watermarked images and forged images for the attacks at 50, 100, and 150 optimization steps (50/100/150).

Method	Attacker Scheduler: DPM (SD 2.1)
	AUC↑
TR+P	0.9924/0.9887/0.9855
GS+P	1.0000/1.0000/1.0000
PRC+P	—
T2SMark+P	1.0000/1.0000/0.9998

Table 12: Results against VAE-Forgery attack on MS-COCO dataset. The attacker optimizes the pixel-space perturbation for 100 steps using the SD 1.4 proxy VAE. Lower AUC and Det. indicate higher robustness to the attack. The best result for each scheme is in **bold**.

Method	Attacker Proxy VAE: SD 1.4		
	Det.↓	Bit Acc.↓	AUC↓
TR	0.77	—	0.9565
TR+P	**0.27**	—	**0.7326**
GS	0.99	0.9359	1.0000
GS+P	**0.02**	**0.6138**	**0.9866**
T2SMark	1.00	0.9814	1.0000
T2SMark+P	**0.02**	**0.5156**	**0.5744**

- Against VAE-Forgery: As presented in Table 12 and Table 13, PGID-F successfully reduces the detection rate of forged instances across all watermarking methods. The AUC and bit accuracy also drop, indicating effective defense against this forgery attack.
- Against Average-Removal: As shown in Table 14, PGID-R successfully restores the watermarks for Tree Ring, indicating our method's potential for defending other perturbation-based removal attacks.

F Evaluation on Larger Test Sets

To further assess the robustness of PGID, we extend our evaluation to 400 images per category for Tree-Ring and Gaussian Shading, quadrupling the original test set size. We follow the same experimental settings as in our main experiments, except for the increased test set size. Tables 15 and 16 report the results against imprint removal and forgery attacks, respectively. For the forgery evaluation, AUC is calculated between watermarked and forged images. The results demonstrate

Table 13: Separability of authentic watermarks against VAE-forgery attack. AUC is calculated between watermarked images and forged images.

Method	Attacker Proxy VAE: SD 1.4
	AUC↑
TR+P	0.9381
GS+P	0.9996
T2SMark+P	0.9995

Table 14: Results against the Average-Removal attack on the Tree Ring. PGID effectively recovers the removed watermarks, as indicated by the restored detection rate and high AUC.

Method	Det.↑	AUC↑
TR	0.10	0.1460
TR+P	**0.97**	**0.9885**

Table 15: Results against imprint removal attacks on 400 images per category. Results are reported for the attack at 50/100/150 optimization steps (Attacker Model: SD 2.1).

Method	Det.↑	Bit Acc.↑	AUC↑
TR+P	0.9975/0.9975/1.0000	—	0.9999/0.9998/0.9999
GS+P	1.0000/1.0000/1.0000	0.9970/0.9980/0.9977	1.0000/1.0000/1.0000

Table 16: Results against imprint forgery attacks on 400 images per category. Results are reported for the attack at 50/100/150 optimization steps (Attacker Model: SD 2.1).

Method	Det.↓	Bit Acc.↓	AUC↑
TR+P	0.0375/0.0650/0.1100	—	0.9912/0.9873/0.9826
GS+P	0.0025/0.0025/0.0325	0.5414/0.5815/0.6133	0.9998/0.9996/0.9990

that PGID maintains strong performance on the larger test set, confirming its effectiveness as the evaluation set size increases.

G Limitations

While PGID significantly enhances the robustness of semantic watermark detection, we acknowledge a few limitations in our current framework. First, the hyperparameter configurations utilized in our evaluations were determined empirically through trial and error, meaning they may not represent the strictly optimal settings. Second, optimal hyperparameter settings might vary depending on the watermarking scheme and model architecture used by the service provider. As mentioned in Section B, watermarking schemes with high sensitivity and low robustness, such as PRC, might require more careful and extensive hyperparameter tuning. Finally, because PGID is fundamentally designed to mitigate perturbations, it currently cannot protect against perturbation-free attacks such as Rinse [17] or the recent next-frame prediction attack [43].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction correctly describe our contributions, which are empirically backed by our experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Discussed in the Limitations part in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Results are based on empirical experiments.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details are in the Experiment and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code will be released upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Included in Experiment and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments are reproducible based on the random seeds we used, so no need for this.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Included in Experiments and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research done in this paper conforms to the NeurIPS Code of Ethics

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Discussed in Abstract, Introduction, and Conclusion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: This paper cites the original owners.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: This paper's core method development does not involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.