Shallow Diffuse: Robust and Invisible Watermarking through Low-Dim Subspaces in Diffusion Models

Wenda Li* **Huijie Zhang*** **Qing Qu**
Department of Electrical Engineering & Computer Science
University of Michigan, Ann Arbor

Abstract

The widespread use of AI-generated content from diffusion models has raised significant concerns regarding misinformation and copyright infringement. Watermarking is a crucial technique for identifying these AI-generated images and preventing their misuse. In this paper, we introduce *Shallow Diffuse*, a new watermarking technique that embeds robust and invisible watermarks into diffusion model outputs. Unlike existing approaches that integrate watermarking throughout the entire diffusion sampling process, *Shallow Diffuse* decouples these steps by leveraging the presence of a low-dimensional subspace in the image generation process. This method ensures that a substantial portion of the watermark lies in the null space of this subspace, effectively separating it from the image generation process. Our theoretical and empirical analyses show that this decoupling strategy greatly enhances the consistency of data generation and the detectability of the watermark. Extensive experiments further validate that *Shallow Diffuse* outperforms existing watermarking methods in terms of consistency.

1 Introduction

Diffusion models [1, 2] have recently become a new dominant family of generative models, powering various commercial applications such as Stable Diffusion [3, 4], DALL-E [5, 6], Imagen [7], Stable Audio [8] and Sora [9]. These models have significantly advanced the capabilities of text-to-image, text-to-audio, text-to-video, and multi-modal generative tasks. However, the widespread usage of AI-generated content from commercial diffusion models on the Internet has raised several serious concerns: (a) AI-generated misinformation presents serious risks to societal stability by spreading unauthorized or harmful narratives on a large scale [10–12]; (b) the memorization of training data by those models [13–17] challenges the originality of the generated content and raises potential copyright infringement issues; (c) iterative training on AI-generated content, known as model collapse [18–22] can degrade the quality and diversity of outputs over time, resulting in repetitive, biased, or low-quality generations that may reinforce misinformation and distortions in the wild Internet.

To deal with these challenges, watermarking is a crucial technique for identifying AI-generated content and mitigating its misuse. Typically, it can be applied in two main scenarios: (a) *the server scenario*, where given an initial random seed, the watermark is embedded into the image during the generation process; and (b) *the user scenario*, where given a generated image, the watermark is injected in a post-processing manner; (as shown in the left two blocks in Figure 2 top). Traditional watermarking methods [23–26] are mainly designed for the user scenario, embedding detectable watermarks directly into images with minimal modification. However, these methods are susceptible to attacks. For example, the watermarks can become undetectable with simple corruptions such as blurring on watermarked images. More recent methods considered the server scenario [27–32], enhancing robustness by integrating watermarking into the sampling process of diffusion models.

*The first and second authors contribute equally to this work.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

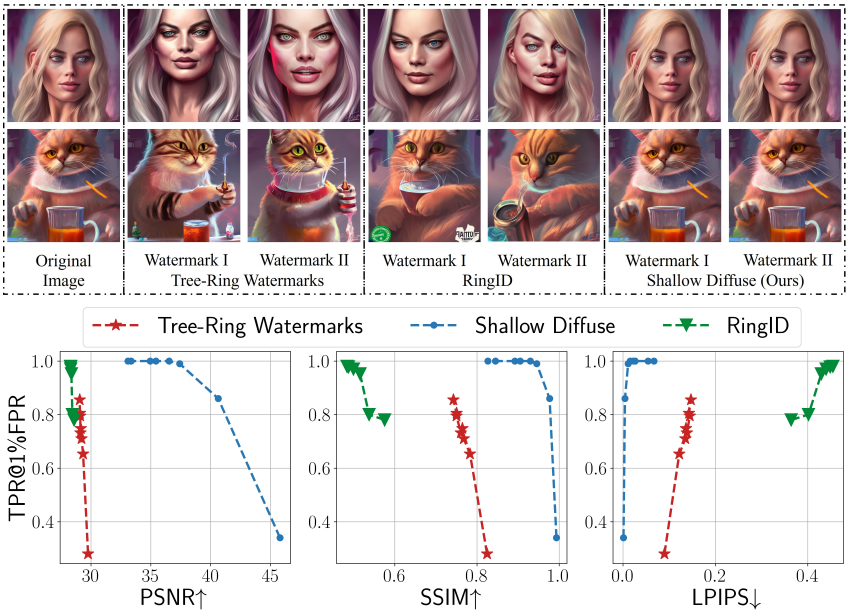


Figure 1: **Comparison between Tree-Ring Watermarks, RingID and Shallow Diffuse.** (Top) On the left are the original images, and on the right are the corresponding watermarked images generated using three techniques: Tree-Ring [29], RingID [31], and Shallow Diffuse. For each technique, we sampled watermarks using two distinct random seeds and obtained the respective watermarked images. (Bottom) Trade-off between consistency (measured by PSNR, SSIM, LPIPS) and robustness (measured by TPR@1%FPR) for Tree-Ring Watermarks, RingID, and Shallow Diffuse.

For example, recent works [29, 31] embed the watermark into the initial random seed in the Fourier domain and then sample an image from the watermarked seed. As illustrated in Figure 1, these methods frequently result in inconsistent watermarked images because they substantially distort the original Gaussian noise distribution. Moreover, since they require access to the initial random seed, it limits their use in the user scenario. To the best of our knowledge, there is no robust and consistent watermarking method suitable for both the server and user scenarios (a more detailed discussion of related works is provided in Appendix B).

To address these limitations, we proposed *Shallow Diffuse*, a robust and consistent watermarking approach that can be employed for both the server and user scenarios. In contrast to prior works [29, 31], which embed watermarks into the initial random seed and tightly couple watermarking with the sampling process, Shallow Diffuse decouples these two steps by exploiting the low-dimensional subspace structure inherent in the generation process of diffusion models [33, 34]. The key insight is that, due to the low dimensionality of the subspace, a significant portion of the watermark will lie in its null space, which effectively separates the watermarking from the sampling process (see Figure 2 for an illustration). Our theoretical and empirical analyses demonstrate that this decoupling strategy significantly improves the consistency of the watermark. Moreover, Shallow Diffuse is flexible for both server and user scenarios, with better consistency as well as independence from the initial random seed.

Our contributions. In summary, our proposed Shallow Diffuse offers several key advantages over existing watermarking techniques [23–32] that we highlight below:

- **Flexibility.** Watermarking via Shallow Diffuse works seamlessly under both server-side and user-side scenarios. In contrast, most of the previous methods only focus on one scenario without an easy extension to the other; see Table 1 and Table 2 for demonstrations.

- **Consistency and robustness.** By decoupling the watermarking from the sampling process, Shallow Diffuse achieves better consistency and comparable robustness. Extensive experiments (Table 1 and Table 2) support our claims, with extra ablation studies in Figure 7a and Figure 7b.

- **Provable guarantees.** The consistency and detectability of our approach are theoretically justified. Assuming a proper low-dimensional image data distribution (see Assumption 1), we rigorously establish bounds for consistency (Theorem 1) and detectability (Theorem 2).

2 Preliminaries

We start by reviewing the basics of diffusion models [1, 2, 35], followed by several key empirical properties that will be used in our approach: the low-rankness and local linearity of the diffusion model [33, 34].

2.1 Preliminaries on Diffusion Models

Basics of diffusion models. In general, diffusion models consist of two processes:

- *The forward diffusion process.* The forward process progressively perturbs the original data $\boldsymbol{x}_0$ to a noisy sample $\boldsymbol{x}_t$ for some integer $t \in [0, T]$ with $T \in \mathbb{Z}$. As in [1], this can be characterized by a conditional Gaussian distribution $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\mathbf{I}_d)$. Particularly, parameters $\{\alpha_t\}_{t=0}^T$ sastify: (*i*) $\alpha_0 = 1$, and thus $p_0 = p_{\text{data}}$, and (*ii*) $\alpha_T = 0$, and thus $p_T = \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$.
- *The reverse sampling process.* To generate a new sample, previous works [1, 35–37] have proposed various methods to approximate the reverse process of diffusion models. Typically, these methods involve estimating the noise $\boldsymbol{\epsilon}_t$ and removing the estimated noise from $\boldsymbol{x}_t$ recursively to obtain an estimate of $\boldsymbol{x}_0$. Specifically, One sampling step of Denoising Diffusion Implicit Models (DDIM) [36] from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t-1}$ can be described as:

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sqrt{\alpha_t}} \right)}_{:= \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \tag{1}$$

  where $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is parameterized by a neural network and trained to predict the noise $\boldsymbol{\epsilon}_t$ at time t. From previous works [38, 39], the first term in Equation (1), defined as $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$, is the *posterior mean predictor* (PMP) that predict the posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$. DDIM could also be applied to a clean sample $\boldsymbol{x}_0$ and generate the corresponding noisy $\boldsymbol{x}_t$ at time t, named DDIM Inversion. One sampling step of DDIM inversion is similar to Equation (1), by mapping from $\boldsymbol{x}_{t-1}$ to $\boldsymbol{x}_t$. For any t_1 and t_2 with $t_2 > t_1$, we denote multi-time steps DDIM operator and its inversion as $\boldsymbol{x}_{t_1} = \texttt{DDIM}(\boldsymbol{x}_{t_2}, t_1)$ and $\boldsymbol{x}_{t_2} = \texttt{DDIM} - \texttt{Inv}(\boldsymbol{x}_{t_1}, t_2)$.

Text-to-image (T2I) diffusion models & classifier-free guidance (CFG). The diffusion model can be generalized from unconditional to T2I [3, 4], where the latter enables controllable image generation $\boldsymbol{x}_0$ guided by a text prompt $\boldsymbol{c}$. In more detail, when training T2I diffusion models, we optimize a conditional denoising function $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})$. For sampling, we employ a technique called *classifier-free guidance* (CFG) [40], which substitutes the unconditional denoiser $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ in Equation (1) with its conditional counterpart $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})$ that can be described as $\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c}) = (1-\eta)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{\varnothing}) + \eta\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, \boldsymbol{c})$.. Here, $\boldsymbol{\varnothing}$ denotes the empty prompt, and $\eta > 0$ denotes the strength for the classifier-free guidance. For simplification, for any t_1 and t_2 with $t_2 > t_1$, we denote multi-time steps CFG operator as $\boldsymbol{x}_{t_1} = \texttt{CFG}(\boldsymbol{x}_{t_2}, t_1, \boldsymbol{c})$. DDIM and DDIM inversion could also be generalized to T2I version, denoted by $\boldsymbol{x}_{t_1} = \texttt{DDIM}(\boldsymbol{x}_{t_2}, t_1, \boldsymbol{c})$ and $\boldsymbol{x}_{t_2} = \texttt{DDIM} - \texttt{Inv}(\boldsymbol{x}_{t_1}, t_2, \boldsymbol{c})$.

2.2 Local Linearity and Intrinsic Low-Dimensionality in PMP

In this work, we leverage two key properties of the PMP $f_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ introduced in Equation (1) for watermarking diffusion models. Parts of these properties have been previously identified in recent papers [33, 41, 42], and have been extensively analyzed in [34]. At a given timestep $t \in [0, T]$, consider the first-order Taylor expansion of the PMP $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda\Delta\boldsymbol{x})$ at the point $\boldsymbol{x}_t$:

$$\boxed{\boldsymbol{l}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \lambda\Delta\boldsymbol{x}) \;:=\; \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \lambda\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \cdot \Delta\boldsymbol{x},} \tag{2}$$

where $\Delta\boldsymbol{x} \in \mathbb{S}^{d-1}$ is a perturbation direction with unit length, $\lambda \in \mathbb{R}$ is the perturbation strength, and $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ denotes the Jacobian of $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$. Within a certain range of noise levels, the learned PMP $\boldsymbol{f}_{\boldsymbol{\theta},t}$ exhibits local linearity, and its Jacobian $\boldsymbol{J}_{\boldsymbol{\theta},t} \in \mathbb{R}^{d\times d}$ is low rank:

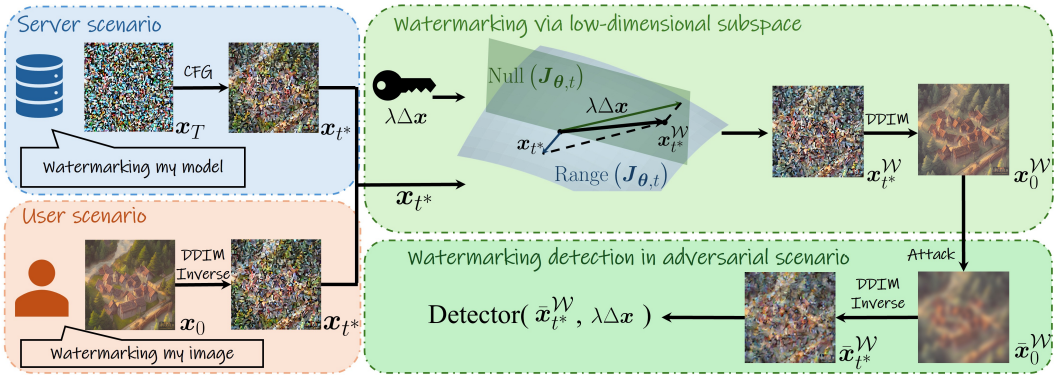


Figure 2: **Overview of Shallow Diffuse for T2I Diffusion Models.** The *server scenario* (top left) illustrates watermark embedding during generation using CFG, while the *user scenario* (bottom left) demonstrates post-generation watermark embedding via DDIM inversion. In both scenarios, the watermark is applied within a *low-dimensional subspace* (top right), where most of the watermark resides in the null space of $\boldsymbol{J}_{\boldsymbol{\theta},t}$ due to its low dimensionality. The *adversarial detection* (bottom right) highlights the watermark's robustness, enabling the detector to retrieve the watermark even under adversarial attacks.

- **Low-rankness of the Jacobian $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$.** As shown in Figure 2(a) of [34], the *rank ratio* for $t \in [0, T]$ *consistently* displays a U-shaped pattern across various network architectures and datasets: (*i*) it is close to 1 near either the pure noise $t = T$ or the clean image $t = 0$, (*ii*) $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ is low-rank (i.e., the numerical rank ratio is below 10^{-2}) for all diffusion models within the range $t \in [0.2T, 0.7T]$.
- **Local linearity of the PMP $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$.** As shown in [34, 43], the mapping $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ exhibits strong linearity across a large portion of the timesteps, i.e., $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda\Delta\boldsymbol{x}) \approx \boldsymbol{l}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \lambda\Delta\boldsymbol{x})$, a property that holds consistently true across different architectures trained on different datasets.

3 Watermarking by Shallow-Diffuse

This section introduces Shallow Diffuse, a training-free watermarking method designed for diffusion models. Building on the benign properties of PMP discussed in Section 2.2, we describe how to inject and detect invisible watermarks in unconditional diffusion models in Section 3.1 and Section 3.2, respectively. Algorithm 1 outlines the overall watermarking method for unconditional diffusion models. In Section 3.3, we generalize our approach to T2I diffusion models as shown in Figure 2.

Algorithm 1 Unconditional Shallow Diffuse

Inject watermark:
Input: original image $\boldsymbol{x}_0$ for the user scenario (initial random seed $\boldsymbol{x}_T$ for the server scenario), watermark $\lambda\Delta\boldsymbol{x}$, embedding timestep t^*,
Output: watermarked image $\boldsymbol{x}_0^{\mathcal{W}}$,
if user scenario **then**
  $\boldsymbol{x}_{t^*} = \texttt{DDIM}-\texttt{Inv}\left(\boldsymbol{x}_0, t^*\right)$
else server scenario
  $\boldsymbol{x}_{t^*} = \texttt{DDIM}\left(\boldsymbol{x}_T, t^*\right)$
end if
$\boldsymbol{x}_{t^*}^{\mathcal{W}} \leftarrow \boldsymbol{x}_{t^*} + \lambda\Delta\boldsymbol{x}$, $\boldsymbol{x}_0^{\mathcal{W}} \leftarrow \texttt{DDIM}\left(\boldsymbol{x}_{t^*}^{\mathcal{W}}, 0\right)$
▷ Embed watermark
Return: $\boldsymbol{x}_0^{\mathcal{W}}$

Detect watermark:
Input: Attacked image $\bar{\boldsymbol{x}}_0^{\mathcal{W}}$, watermark $\lambda\Delta\boldsymbol{x}$, embedding timestep t^*,
Output: Distance score η,
$\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}} \leftarrow \texttt{DDIM}-\texttt{Inv}\left(\bar{\boldsymbol{x}}_0^{\mathcal{W}}, t^*\right)$
$\eta = \texttt{Detector}\left(\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}}, \lambda\Delta\boldsymbol{x}\right)$
Return: η

3.1 Injecting Invisible Watermarks

Consider an unconditional diffusion model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ as introduced in Section 2.1. Instead of injecting the watermark $\Delta\boldsymbol{x}$ in the initial noise, we inject it in a particular timestep $t^* \in [0, T]$ with

$$\boldsymbol{x}_{t^*}^{\mathcal{W}} = \boldsymbol{x}_{t^*} + \lambda\Delta\boldsymbol{x}, \tag{3}$$

where $\lambda \in \mathbb{R}$ is the watermarking strength, $\boldsymbol{x}_{t^*} = \texttt{DDIM}-\texttt{Inv}\left(\boldsymbol{x}_0, t^*\right)$ under the user scenario and $\boldsymbol{x}_{t^*} = \texttt{DDIM}\left(\boldsymbol{x}_T, t^*\right)$ under the server scenario. Based upon Section 2.2, we choose the timestep t^* so that the Jacobian of the PMP $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*}) = \nabla_{\boldsymbol{x}_t}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*})$ is *low-rank*. Moreover, based

upon the linearity of PMP discussed in Section 2.2, we approximately have

$$\begin{aligned} \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*}^{\mathcal{W}}) &= \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*}) + \lambda \underbrace{\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*})\Delta\boldsymbol{x}}_{\approx \mathbf{0}} \\ &\approx \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*}), \end{aligned} \tag{4}$$

where the watermark $\Delta\boldsymbol{x}$ is designed to span the entire space $\mathbb{R}^d$ *uniformly*; a more detailed discussion on the pattern design of $\Delta\boldsymbol{x}$ is provided in Section 3.2. The key intuition for Equation (4) to hold is that, when $r_{t^*} = \mathrm{rank}(\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*}))$ is low, a significant proportion of $\lambda\Delta\boldsymbol{x}$ lies in the *null space* of $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*})$, so that $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*})\Delta\boldsymbol{x} \approx \mathbf{0}$.

Therefore, the selection of t^* is based on the requirement that $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t^*)$ is locally linear and that the rank of its Jacobian satisfies $r_t^* \ll d$. In practice, we choose $t^* = 0.3T$ based on results from the ablation study in Appendix C.5. As a result, the injection in Equation (4) preserves better consistency without changing the predicted $\boldsymbol{x}_0$. In the meanwhile, it remains highly robust because any attack on $\boldsymbol{x}_0$ would remain disentangled from the watermark, so that $\lambda\Delta\boldsymbol{x}$ remains detectable.

In practice we employ the DDIM method instead of PMP for sampling high-quality images, but the above intuition still carries over to DDIM. From Equation (1), when we inject the watermark $\Delta\boldsymbol{x}$ into $\boldsymbol{x}_t^*$ as given in Equation (3), we know that

$$\begin{aligned} \boldsymbol{x}_{t^*-1}^{\mathcal{W}} &= \texttt{DDIM}(\boldsymbol{x}_{t^*}^{\mathcal{W}}, t^* - 1) \\ &\approx \sqrt{\alpha_{t^*-1}}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*}) + \frac{\sqrt{1-\alpha_{t^*-1}}}{\sqrt{1-\alpha_{t^*}}}\left(\boldsymbol{x}_{t^*} + \lambda\Delta\boldsymbol{x} - \sqrt{\alpha_{t^*}}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_{t^*})\right), \end{aligned} \tag{5}$$

where the approximation follows from Equation (4). This implies that the watermark $\lambda\Delta\boldsymbol{x}$ is embedded into the DDIM sampling process entirely through the second term of Equation (5) and it decouples from the first term, which predicts $\boldsymbol{x}_0$. Therefore, similar to our analysis for PMP, the first term in Equation (5) maintains the consistency of data generation, whereas the difference in the second term, highlighted in blue, serves as a key feature for watermark detection, which we will discuss next. In Section 4, we provide rigorous proofs validating the consistency and detectability of our approach.

3.2 Watermark Design and Detection

Second, building on the watermark injection method described in Section 3.1, we discuss the design of the watermark pattern and the techniques for effective detection.

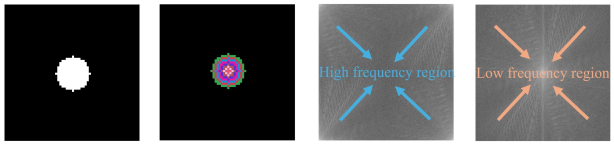
Watermark pattern design. Building on the method proposed by [29], we inject the watermark in the frequency domain to enhance robustness against adversarial attacks. Specifically, we adapt this approach by defining a watermark $\lambda\Delta\boldsymbol{x}$ for the input $\boldsymbol{x}_{t^*}$ at timestep t^* as follows:

$$\lambda\Delta\boldsymbol{x} \;:=\; \texttt{DFT} - \texttt{Inv}\left(\texttt{DFT}\left(\boldsymbol{x}_{t^*}\right) \odot (1 - \boldsymbol{M}) + \boldsymbol{W} \odot \boldsymbol{M}\right) - \boldsymbol{x}_{t^*}, \tag{6}$$

where the Hadamard product $\odot$ denotes the element-wise multiplication. Additionally, we have the following for Equation (6):

- **Transformation into the frequency domain.** Let $\texttt{DFT}(\cdot)$ and $\texttt{DFT} - \texttt{Inv}(\cdot)$ denote the forward and inverse Discrete Fourier Transform (DFT) operators, respectively. As shown in Equation (6), we first apply $\texttt{DFT}(\cdot)$ to transform $\boldsymbol{x}_{t^*}$ into the frequency domain, where the watermark is introduced via a mask. Finally, the modified input is transformed back into the pixel domain using $\texttt{DFT} - \texttt{Inv}(\cdot)$.
- **The mask and key of watermarks.** $\boldsymbol{M}$ is the mask used to apply the watermark in the frequency domain, as shown in the top-left of Figure 3, and $\boldsymbol{W}$ denotes the key of the watermark. Typically, the mask $\mathbf{M}$ is circular, with the white area representing 1 and the black area representing 0 in Figure 3. The mask is used to modify specific frequency bands of the image. Specifically, circular mask M has a radius of 8. In the following, we discuss the design of $\boldsymbol{M}$ and $\boldsymbol{W}$ in detail.

In contrast to prior methods [29, 31], which design the mask $\boldsymbol{M}$ to modify the **low-frequency components** of the initial noise input, we construct $\boldsymbol{M}$ to target the **high-frequency components** of the image. While modifying low-frequency components is effective due to the concentration of image energy in those bands, such approaches often introduce significant visual distortion when



Mask $\boldsymbol{M}$ | $\boldsymbol{W} \odot \boldsymbol{M}$ | DFT w/o shift | DFT with shift

Figure 3: **Visualization of Watermark Patterns**. The left two images show the circular mask $\boldsymbol{M}$ and the key within the mask $\boldsymbol{M} \odot \boldsymbol{W}$, where the key $\boldsymbol{W}$ consists of multiple rings and each sampled from the Gaussian distribution. The right two images illustrate the low- and high-frequency regions applying DFT, both before and after centering the zero frequency.

watermarks are embedded (see Figure 1 for illustration). In contrast, as shown in Figure 3, our method introduces minimal distortion by operating on the high-frequency components, which correspond to finer details and inherently contain less energy. This effect is further amplified in our case, as we apply the perturbation to $\boldsymbol{x}_{t^*}$, which is closer to the clean image $\boldsymbol{x}_0$, rather than to the initial noise used in [29, 31]. To isolate the high-frequency components, we apply the DFT without shifting and centering the zero-frequency component, as illustrated in the bottom-left of Figure 3.

In designing the key $\boldsymbol{W}$, we follow [29]. The key $\boldsymbol{W}$ is composed of multi-rings and each ring has the same value drawn from Gaussian distribution; see the top-right of Figure 3 for an illustration. Further ablation studies on the choice of $\boldsymbol{M}$, $\boldsymbol{W}$, and the effects of selecting low-frequency versus high-frequency regions for watermarking can be found in Table 8.

Watermark detection. During watermark detection, suppose we are given a watermarked image $\bar{\boldsymbol{x}}_0^{\mathcal{W}}$ with certain corruptions, we apply DDIM Inversion to recover the watermarked image at timestep t^*, denoted as $\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}} = \texttt{DDIM}-\texttt{Inv}\left(\bar{\boldsymbol{x}}_0^{\mathcal{W}}, t\right)$. To detect the watermark, following [27, 29], the $\texttt{Detector}(\cdot)$ in Algorithm 1 computes the following p-value:

$$\eta = \frac{\texttt{sum}(\boldsymbol{M}) \cdot ||\boldsymbol{M} \odot \boldsymbol{W} - \boldsymbol{M} \odot \texttt{DFT}\left(\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}}\right)||_F^2}{||\boldsymbol{M} \odot \texttt{DFT}\left(\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}}\right)||_F^2}, \tag{7}$$

where $\texttt{sum}(\cdot)$ is the summation of all elements of the matrix. Ideally, if $\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}}$ is a watermarked image, $\boldsymbol{M} \odot \boldsymbol{W} = \boldsymbol{M} \odot \texttt{DFT}\left(\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}}\right)$ and $\eta = 0$. When $\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}}$ is a non-watermarked image, $\boldsymbol{M} \odot \boldsymbol{W} \neq \boldsymbol{M} \odot \texttt{DFT}\left(\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}}\right)$ and $\eta > 0$. By selecting a threshold η_0, non-watermarked images satisfy $\eta > \eta_0$, while watermarked images satisfy $\eta < \eta_0$. The theoretical derivation of the p-value η could be found in [27].

3.3 Extension to Text-to-Image (T2I) Diffusion Models

So far, our discussion has focused exclusively on unconditional diffusion models. Next, we show how our approach can be readily extended to T2I diffusion models, which are widely used in practice. Specifically, Figure 2 provides an overview of our method for T2I diffusion models, which can be flexibly applied to both server and user scenarios:

- **Watermark injection.** Shallow Diffuse embeds watermarks into the noise corrupted image $\boldsymbol{x}_{t^*}$ at a specific timestep $t^* = 0.3T$. In the **server scenario**, given $\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ and prompt $\boldsymbol{c}$, we calculate $\boldsymbol{x}_{t^*} = \texttt{CFG}\left(\boldsymbol{x}_T, t^*, \boldsymbol{c}\right)$. In the **user scenario**, given the generated image $\boldsymbol{x}_0$, we compute $\boldsymbol{x}_{t^*} = \texttt{DDIM}-\texttt{Inv}\left(\boldsymbol{x}_0, t^*, \varnothing\right)$, using an empty prompt $\varnothing$. Next, similar to Section 3.1, we apply DDIM to obtain the watermarked image $\boldsymbol{x}_0^{\mathcal{W}} = \texttt{DDIM}\left(\boldsymbol{x}_{t^*}^{\mathcal{W}}, 0, \varnothing\right)$.
- **Watermark detection.** During watermark detection, suppose we are given a watermarked image $\bar{\boldsymbol{x}}_0^{\mathcal{W}}$ with certain corruptions, we apply the DDIM Inversion to recover the watermarked image at timestep t^*, denoted as $\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}} = \texttt{DDIM}-\texttt{Inv}\left(\bar{\boldsymbol{x}}_0^{\mathcal{W}}, t^*, \varnothing\right)$. We detect the watermark $\Delta\boldsymbol{x}$ in $\bar{\boldsymbol{x}}_{t^*}^{\mathcal{W}}$ by calculating η in Equation (7), with detail explained in Section 3.2.

4 Theoretical Justification

In this section, we provide theoretical justifications for the consistency and the detectability of Shallow Diffuse for unconditional diffusion models. We begin by making the following assumptions on the watermark and the diffusion process.

Assumption 1. *Suppose the following holds for the PMP $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ introduced in Equation* (1)*:*

- ***Linearity:*** *For any t and $\Delta\boldsymbol{x} \in \mathbb{S}^{d-1}$, we always have $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t+\lambda\Delta\boldsymbol{x}) = \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)+\lambda\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}$.*
- ***L-Lipschitz continuous:*** *we assume that $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x})$ is L-Lipschiz continuous $||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x})||_2 \leq L, \forall \boldsymbol{x} \in \mathbb{R}^d, t \in [0,T]$*

It should be noted that these assumptions are mild. The L-Lipschitz continuity is a common assumption for diffusion model analysis [44–49]. The approximated linearity have been shown in [34] with the assumption of data distribution to follow a mixture of low-rank Gaussians. For the ease of analysis, we assume exact linearity, but it can be generalized to the approximate linear case with extra perturbation analysis.

Now consider injecting a watermark $\lambda\Delta\boldsymbol{x}$ in Equation (3), where $\lambda > 0$ is a scaling factor and $\Delta\boldsymbol{x}$ is a *random* vector uniformly distributed on the unit hypersphere $\mathbb{S}^{d-1}$, i.e., $\Delta\boldsymbol{x} \sim \mathrm{U}(\mathbb{S}^{d-1})$. Then the following hold for $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$.

Theorem 1 (Consistency of the watermarks)**.** *Suppose Assumption 1 holds and $\Delta\boldsymbol{x} \sim \mathrm{U}(\mathbb{S}^{d-1})$. Define $\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} \coloneqq \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda\Delta\boldsymbol{x})$, $\hat{\boldsymbol{x}}_{0,t} \coloneqq \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$. Then the ℓ_2-norm distance between $\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}}$ and $\hat{\boldsymbol{x}}_{0,t}$ is bounded by:*

$$||\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t}||_2 \leq \lambda L h(r_t), \tag{8}$$

with probability at least $1 - r_t^{-1}$. Here, $h(r_t) = \sqrt{\frac{r_t}{d} + \sqrt{\frac{18\pi^3}{d-2}\log(2r_t)}}$.

Theorem 1 guarantees that injecting the watermark $\lambda\Delta\boldsymbol{x}$ would only change the estimation by an amount of $\lambda L h(r_t)$ with a constant probability, where $h(r_t)$ ***only*** **depends on the rank of the Jacobian r_t ($r_t \ll d$) rather than the ambient dimension** d. Since r_t is small, Equation (8) implies that the change in the prediction would be small. Given the relationship between PMP and DDIM in equation 1, the consistency also applies to practical use. Moreover, in the following, we show that the injected watermark remains detectable based on the second term in Equation (5).

Theorem 2 (Detectability of the watermark)**.** *Suppose Assumption 1 holds and $\Delta\boldsymbol{x} \sim \mathrm{U}(\mathbb{S}^{d-1})$. With $\boldsymbol{x}_t^{\mathcal{W}}$ given in Equation* (3)*, define $\boldsymbol{x}_{t-1}^{\mathcal{W}} = \texttt{DDIM}\left(\boldsymbol{x}_t^{\mathcal{W}}, t-1\right)$ and $\bar{\boldsymbol{x}}_t^{\mathcal{W}} = \texttt{DDIM}-\texttt{Inv}\left(\boldsymbol{x}_{t-1}^{\mathcal{W}}, t\right)$. The ℓ_2-norm distance between $\tilde{\boldsymbol{x}}_t^{\mathcal{W}}$ and $\boldsymbol{x}_t^{\mathcal{W}}$ can be bounded by:*

$$||\bar{\boldsymbol{x}}_t^{\mathcal{W}} - \boldsymbol{x}_t^{\mathcal{W}}||_2 \leq \lambda L h(\max\{r_{t-1}, r_t\})[-g(\alpha_t, \alpha_{t-1}) + g(\alpha_{t-1}, \alpha_t)(1 - Lg(\alpha_t, \alpha_{t-1}))] \tag{9}$$

with probability at least $1 - r_t^{-1} - r_{t-1}^{-1}$. Here, $g(x,y) \coloneqq \frac{\sqrt{1-y}\sqrt{x}-\sqrt{1-x}\sqrt{y}}{\sqrt{1-x}}$, $\forall x, y \in (0,1)$. $h(r_t) = \sqrt{\frac{r_t}{d} + \sqrt{\frac{18\pi^3}{d-2}\log(2r_t)}}$.

Similarly, the term $h(\max\{r_{t-1}, r_t\})$ is small because it only depends on the rank of the Jacobian r_t or r_{t-1} ($r_{t-1}, r_t \ll d$) rather than the ambient dimension d. Additionally, the term $-g(\alpha_t, \alpha_{t-1}) + g(\alpha_{t-1}, \alpha_t)(1 - Lg(\alpha_t, \alpha_{t-1}))$ is also a small number based on the design of α_t for variance preserving (VP) noise scheduler [1]. Together, this implies that the difference between $\bar{\boldsymbol{x}}_t^{\mathcal{W}}$ and $\boldsymbol{x}_t^{\mathcal{W}}$ is small and $\boldsymbol{x}_t^{\mathcal{W}}$ could be recovered by $\bar{\boldsymbol{x}}_t^{\mathcal{W}}$ from one-step DDIM. Therefore, Theorem 2 implies that the injected watermark can be detected with high probability.

5 Experiments

In this section, we present a comprehensive set of experiments to demonstrate the robustness and consistency of *Shallow-Diffuse* across various datasets. We begin by highlighting its performance in terms of robustness and consistency in both the server scenario (Section 5.1) and the user scenario (Section 5.2). We further explore the trade-off between robustness and consistency in Section 5.3. Lastly, we provide extra *multi-key identification* experiments in Appendix C.2 and ablation studies on watermark pattern design (Appendix C.3), watermarking embedded channel (Appendix C.4), watermark injecting timestep t (Appendix C.5) and inference steps (Appendix C.6).

Table 1: **Generation quality, consistency and watermark robustness under the server scenario. Bold** indicates the best overall performance; Underline denotes the best among diffusion-based methods.

Method	Generation Quality		Generation Consistency			Watermark Robustness (TPR@1%FPR↑)				
	CLIP-Score ↑	FID ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Clean	Distortion	Regeneration	Adversarial	Average
SD w/o WM	0.3669	25.56	-	-	-	-	-	-	-	-
DwtDct	0.3641	25.73	40.32	0.98	**0.01**	0.85	0.35	0.01	0.42	0.22
DwtDctSvd	0.3629	26.00	40.19	0.98	**0.01**	**1.00**	0.74	0.07	0.01	0.37
RivaGAN	0.3628	24.60	**40.45**	**0.99**	**0.01**	0.99	0.88	0.05	**0.82**	0.54
Stegastamp	0.3410	**24.59**	26.70	0.85	0.08	**1.00**	0.99	0.48	0.05	0.66
Stable Signature	0.3622	30.86	32.43	0.95	0.02	**1.00**	0.59	0.19	0.99	0.48
Tree-Ring	0.3645	25.82	16.61	0.64	0.31	**1.00**	0.88	0.87	0.06	0.77
RingID	0.3637	27.13	14.27	0.51	0.42	**1.00**	**1.00**	**1.00**	0.33	0.91
Gaussian Shading	0.3663	26.17	11.04	0.48	0.54	**1.00**	**1.00**	**1.00**	0.47	**0.93**
Shallow Diffuse	**0.3669**	25.60	35.49	0.96	0.02	**1.00**	**1.00**	0.98	0.54	**0.93**

Comparison baselines. For the server scenario, we select the following non-diffusion-based methods: DWtDct [23], DwtDctSvd [23], RivaGAN [50], StegaStamp [51]; and diffusion-based methods: Stable Signature [28], Tree-Ring Watermarks [29], RingID [31], and Gaussian Shading [30]. In the user scenario, we adopt the same baseline methods, except for Stable Signature, as this method are not suitable for this setting.

Evaluation datasets. We use Stable Diffusion 2-1-base [3] as the underlying model for our experiments, applying Shallow diffusion within its latent space. For the server scenario (Section 5.1), all diffusion-based methods are based on the same Stable Diffusion, with the original images $\boldsymbol{x}_0$ generated from identical initial seeds $\boldsymbol{x}_T$. Non-diffusion methods are applied to these same original images $\boldsymbol{x}_0$ in a post-watermarking process. A total of 5000 original images are generated for evaluation in this scenario. For the user scenario (Section 5.2), we utilize the MS-COCO [52], and DiffusionDB datasets [53]. The first one is a real-world dataset, while DiffusionDB is a collection of diffusion model-generated images. From each dataset, we select 500 images for evaluation. For the remaining experiments in Section 5.3 and Appendix C, we use the server scenario and sample 100 images for evaluation.

Evaluation metrics. To evaluate image consistency, we use peak signal-to-noise ratio (PSNR) [54], structural similarity index measure (SSIM) [55], and Learned Perceptual Image Patch Similarity (LPIPS) [56], comparing watermarked images to their original counterparts. In the server scenario, we also assess the generation quality of the watermarked images using Contrastive Language-Image Pretraining Score (CLIP-Score) [57] and Fréchet Inception Distance (FID) [58]. To evaluate robustness, we plot the true positive rate (TPR) against the false positive rate (FPR) for the receiver operating characteristic (ROC) curve. We use the area under the curve (AUC) and TPR when FPR = 0.01 (TPR $@1\%$ FPR) as robustness metrics.

Attacks. Robustness is comprehensively evaluated both under clean conditions (no attacks) and with **15 types of attacks**. Following [59], we categorized them into three groups, including: distortion attack (JPEG compression, Gaussian blurring, Gaussian noise, color jitter, resize and restore, random drop, median blurring), regeneration attack (diffusion purification [60], VAE-based image compression models [61, 62], stable diffusion-based image regeneration [63], 2 times and 4 times rinsing regenerations [59]) and adversarial attack (black-box and grey-box averaging attack [64]). Here, we report only the TPR at 1% FPR for the average robustness across each group and all attacks. Detailed settings and full experiment results of these attacks are provided in Appendix C.1.

5.1 Server Scenario Consistency and Robustness

Table 1 compares the performance of Shallow Diffuse with other methods in the server scenario. For reference, we also apply stable diffusion to generate images from the same random seeds, without adding watermarks (referred to as "SD w/o WM" in Table 1). In terms of generation quality, Shallow Diffuse achieves the best FID and CLIP scores among all diffusion-based methods. It also demonstrates superior generation consistency, achieving the highest PSNR, SSIM, and LPIPS scores.

Table 2: **Generation consistency and watermark robustness under the user scenario. Bold** indicates the best overall performance; Underline denotes the best among diffusion-based methods.

	Method	Generation Consistency			Watermark Robustness (AUC ↑/TPR@1%FPR↑)				
		PSNR ↑	SSIM ↑	LPIPS ↓	Clean	Distortion	Regeneration	Adversarial	Average
COCO	SD w/o WM	32.28	0.78	0.06	-	-	-	-	-
	DwtDct	37.88	0.97	**0.02**	0.83	0.54	0.00	0.82	0.36
	DwtDctSvd	38.06	**0.98**	**0.02**	**1.00**	0.76	0.06	0.00	0.38
	RivaGAN	**40.57**	**0.98**	0.04	**1.00**	0.93	0.05	**1.00**	0.59
	Stegastamp	31.88	0.86	0.08	**1.00**	0.97	0.47	0.26	0.68
	Gaussian Shading	10.17	0.23	0.65	**1.00**	0.99	**1.00**	0.47	0.92
	Tree-Ring	28.22	0.57	0.41	**1.00**	0.90	0.95	0.31	0.84
	RingID	12.21	0.38	0.58	**1.00**	0.98	**1.00**	0.79	**0.96**
	Shallow Diffuse	32.11	0.84	0.05	**1.00**	**1.00**	0.96	0.62	0.93
DiffusionDB	SD w/o WM	33.42	0.85	0.03	-	-	-	-	-
	DwtDct	37.77	0.96	**0.02**	0.76	0.34	0.01	0.78	0.27
	DwtDctSvd	37.84	0.97	**0.02**	**1.00**	0.74	0.04	0.00	0.36
	RivaGAN	**40.6**	**0.98**	0.04	0.98	0.88	0.04	**0.98**	0.56
	Stegastamp	32.03	0.85	0.08	**1.00**	0.96	0.46	0.26	0.67
	Gaussian Shading	10.61	0.27	0.63	**1.00**	0.99	**1.00**	0.46	0.92
	Tree-Ring	28.3	0.62	0.29	**1.00**	0.81	0.87	0.26	0.76
	RingID	12.53	0.45	0.53	**1.00**	0.99	**1.00**	0.79	**0.97**
	Shallow Diffuse	33.07	0.89	0.03	**1.00**	**1.00**	0.93	0.59	0.92

Regarding robustness, Shallow Diffuse performs comparably to Gaussian Shading and RingID, while outperforming the remaining methods. Although Gaussian Shading and RingID show similar levels of generation quality and robustness in the server scenario, their poor consistency makes them less suitable for the user scenario.

5.2 User Scenario Consistency and Robustness

Under the user scenario, Table 2 presents a comparison of Shallow Diffuse against other methods. In terms of consistency, Shallow Diffuse outperforms all other diffusion-based approaches. To measure the upper bound of diffusion-based methods, we apply stable diffusion with $\hat{\boldsymbol{x}}_0 = \texttt{DDIM}(\texttt{DDIM} - \texttt{Inv}(\boldsymbol{x}_0, t, \varnothing), 0, \varnothing)$, and measure the data consistency between $\hat{\boldsymbol{x}}_0$ and $\boldsymbol{x}_0$ (denoted in SD w/o WM in Table 2). The upper bound is constrained by errors introduced through DDIM inversion, and Shallow Diffuse comes the closest to reaching this limit. For non-diffusion-based methods, which are not affected by DDIM inversion errors, better image consistency is achievable. However, as visualized in Figure 5, Shallow Diffuse also demonstrates strong generation consistency. In terms of robustness, Shallow Diffuse performs comparably to RingID and Gaussian shading, while outperforming all other methods across both datasets. Notably, RingID and Gaussian achieve high robustness at the sacrifice of poor generation consistency (see Table 2 and Figure 5). In contrast, Shallow Diffuse is the only method that balances strong generation consistency with high watermark robustness, making it suitable for both user and server scenarios.

5.3 Trade-off between Consistency and Robustness

Figure 1 bottom illustrates the trade-off between consistency and robustness [2] for Shallow Diffuse and other baselines. As the radius of $\boldsymbol{M}$ increases, the watermark intensity λ also increases, reducing image consistency but improving robustness. By adjusting the radius of $\boldsymbol{M}$, we plot the trade-off using PSNR, SSIM, and LPIPS against TPR@1%FPR. From Figure 1 bottom, curve of Shallow Diffuse is consistently above the curve of Tree-Ring Watermarks and RingID, demonstrating Shallow Diffuse's better consistency at the same level of robustness.

6 Conclusion

We proposed Shallow Diffuse, a novel and flexible watermarking technique that operates seamlessly in both server-side and user-side scenarios. By decoupling the watermark from the sampling process, Shallow Diffuse achieves enhanced robustness and greater consistency. Our theoretical analysis demonstrates both the consistency and detectability of the watermarks. Extensive experiments further validate the superiority of Shallow Diffuse over existing approaches.

[2] In this experiment, we evaluate robustness against distortion attacks.

References

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

[2] 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.

[3] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[4] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[5] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[6] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

[7] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[8] Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Long-form music generation with latent diffusion. *arXiv preprint arXiv:2404.10301*, 2024.

[9] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.

[10] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.

[11] Josh A Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, and Katerina Sedova. Generative language models and automated influence operations: Emerging threats and potential mitigations. *arXiv preprint arXiv:2301.04246*, 2023.

[12] Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitzoff, Bobby Filar, et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018.

[13] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*, 2023.

[14] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

[15] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[16] Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023.

[17] Benjamin J Zhang, Siting Liu, Wuchen Li, Markos A Katsoulakis, and Stanley J Osher. Wasserstein proximal operators describe score-based generative models and resolve memorization. *arXiv preprint arXiv:2402.06162*, 2024.

[18] Shi Fu, Sen Zhang, Yingjie Wang, Xinmei Tian, and Dacheng Tao. Towards theoretical understandings of self-consuming generative models. In *Forty-first International Conference on Machine Learning*, 2024.

[19] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go MAD. In *The Twelfth International Conference on Learning Representations*, 2024.

[20] Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. A tale of tails: Model collapse as a change of scaling laws. In *Forty-first International Conference on Machine Learning*, 2024.

[21] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

[22] Elizabeth Gibney. Ai models fed ai-generated data quickly spew nonsense. *Nature*, 632(8023):18–19, 2024.

[23] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007.

[24] Vassilios Solachidis and Loannis Pitas. Circularly symmetric watermark embedding in 2-d dft domain. *IEEE transactions on image processing*, 10(11):1741–1753, 2001.

[25] Chin-Chen Chang, Piyu Tsai, and Chia-Chen Lin. Svd-based digital image watermarking scheme. *Pattern Recognition Letters*, 26(10):1577–1586, 2005.

[26] Junxiu Liu, Jiadong Huang, Yuling Luo, Lvchen Cao, Su Yang, Duqu Wei, and Ronglong Zhou. An optimized image watermarking method based on hd and svd in dwt domain. *IEEE Access*, 7:80849–80860, 2019.

[27] Lijun Zhang, Xiao Liu, Antoni Viros Martin, Cindy Xiong Bearfield, Yuriy Brun, and Hui Guan. Robust image watermarking using stable diffusion, 2024.

[28] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023.

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

[30] 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*, pages 12162–12171, 2024.

[31] Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. *arXiv preprint arXiv:2404.14055*, 2024.

[32] Huayang Huang, Yu Wu, and Qian Wang. ROBIN: Robust and invisible watermarks for diffusion models with adversarial optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[33] Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint arXiv:2409.02426*, 2024.

[34] Siyi Chen, Zhang Huijie, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. Exploring low-dimensional subspaces in diffusion models for controllable image editing. In *Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS2024)*, 2024.

[35] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

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

[37] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[38] Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Forty-first International Conference on Machine Learning*, 2024.

[39] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

[40] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[41] Hila Manor and Tomer Michaeli. Zero-shot unsupervised and text-based audio editing using DDPM inversion. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 34603–34629. PMLR, 21–27 Jul 2024.

[42] Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. In *The Twelfth International Conference on Learning Representations*, 2024.

[43] Xiang Li, Yixiang Dai, and Qing Qu. Understanding generalizability of diffusion models requires rethinking the hidden gaussian structure. *Advances in neural information processing systems*, 37:57499–57538, 2024.

[44] Adam Block, Youssef Mroueh, and Alexander Rakhlin. Generative modeling with denoising auto-encoders and langevin sampling. *arXiv preprint arXiv:2002.00107*, 2020.

[45] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. *Advances in Neural Information Processing Systems*, 35:22870–22882, 2022.

[46] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023.

[47] Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pages 4735–4763. PMLR, 2023.

[48] Zhenyu Zhu, Francesco Locatello, and Volkan Cevher. Sample complexity bounds for score-matching: Causal discovery and generative modeling. *Advances in Neural Information Processing Systems*, 36:3325–3337, 2023.

[49] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pages 4672–4712. PMLR, 2023.

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

[51] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2117–2126, 2020.

[52] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[53] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022.

[54] Bernd Jähne. *Digital image processing*. Springer Science & Business Media, 2005.

[55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[56] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[58] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[59] 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, 2024.

[60] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022.

[61] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948, 2020.

[62] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[63] Xuandong Zhao, Kexun Zhang, Yu-Xiang Wang, and Lei Li. Generative autoencoders as watermark attackers: Analyses of vulnerabilities and threats. 2023.

[64] Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Can simple averaging defeat modern watermarks? *Advances in Neural Information Processing Systems*, 37:56644–56673, 2024.

[65] Ali Al-Haj. Combined dwt-dct digital image watermarking. *Journal of computer science*, 3(9):740–746, 2007.

[66] KA Navas, Mathews Cheriyan Ajay, M Lekshmi, Tampy S Archana, and M Sasikumar. Dwt-dct-svd based watermarking. In *2008 3rd international conference on communication systems software and middleware and workshops (COMSWARE'08)*, pages 271–274. IEEE, 2008.

[67] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. Redmark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 146:113157, 2020.

[68] Jae-Eun Lee, Young-Ho Seo, and Dong-Wook Kim. Convolutional neural network-based digital image watermarking adaptive to the resolution of image and watermark. *Applied Sciences*, 10(19):6854, 2020.

[69] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *European Conference on Computer Vision*, 2018.

[70] Gabriel Loaiza-Ganem, Brendan Leigh Ross, Rasa Hosseinzadeh, Anthony L. Caterini, and Jesse C. Cresswell. Deep generative models through the lens of the manifold hypothesis: A survey and new connections. *Transactions on Machine Learning Research*, 2024. Survey Certification, Expert Certification.

[71] Jan Pawel Stanczuk, Georgios Batzolis, Teo Deveney, and Carola-Bibiane Schönlieb. Diffusion models encode the intrinsic dimension of data manifolds. In *Forty-first International Conference on Machine Learning*, 2024.

[72] Hamidreza Kamkari, Brendan Leigh Ross, Rasa Hosseinzadeh, Jesse C Cresswell, and Gabriel Loaiza-Ganem. A geometric view of data complexity: Efficient local intrinsic dimension estimation with diffusion models. In *Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS2024)*, 2024.

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

[74] Sandu Popescu, Anthony J Short, and Andreas Winter. Entanglement and the foundations of statistical mechanics. *Nature Physics*, 2(11):754–758, 2006.

A Impact Statement

In this work, we introduce Shallow Diffuse, a training-free watermarking technique that hides a high-frequency signal in the low-dimensional latent subspaces of diffusion models, enabling invisible yet reliably detectable attribution for both server-side text-to-image generation and user-side post-processing. As synthetic and authentic images flooding the internet, establishing verifiable provenance is essential for copyright protection, misinformation mitigation, and scientific reproducibility. Our method preserves perceptual quality, withstands a wide range of image attacks, and requires no model retraining, making it practical for deployment. We further provide theoretical guarantees on imperceptibility and watermark recoverability, grounded in the low-rank structure of the diffusion latent space. We believe our work contributes to the development of trustworthy generative models and can inform future standards for media authentication, digital content tracking, and responsible AI deployment. While our technique could potentially be repurposed for covert signaling, we emphasize that our goal is to enhance transparency and accountability in generative AI. We encourage responsible use of this research in line with ethical guidelines and broader societal interests.

B Related Work

B.1 Image Watermarking

Image watermarking has long been a crucial method for protecting intellectual property in computer vision [23–26]. Traditional techniques primarily focus on user-side watermarking, where watermarks are embedded into images post-generation. These methods [65, 66] typically operate in the frequency domain to ensure the watermarks are imperceptible. However, such watermarks remain vulnerable to adversarial attacks and can become undetectable after applying simple image manipulations like blurring.

Early deep learning-based approaches to watermarking [27, 28, 67–69] leveraged neural networks to embed watermarks. While these methods improved robustness and imperceptibility, they often suffer from high computational costs during fine-tuning and lack flexibility. Each new watermark requires additional fine-tuning or retraining, limiting their practicality.

More recently, diffusion model-based watermarking techniques have gained attraction due to their ability to seamlessly integrate watermarks during the generative process without incurring extra computational costs. Techniques such as [29–31] embed watermarks directly into the initial noise and retrieve the watermark by reversing the diffusion process. These methods enhance robustness and invisibility but are typically restricted to server-side watermarking, requiring access to the initial random seed. Moreover, the watermarks introduced by [29, 31] significantly alter the data distribution, leading to variance towards watermarks in generated outputs (as shown in Figure 1). Recent work [32] proposes embedding the watermark at an intermediate time step using adversarial optimization.

In contrast to [29, 31], our proposed shallow diffuse disentangles the watermark embedding from the generation process by leveraging the high-dimensional null space. This approach significantly improves watermark consistency while maintaining robustness. Furthermore, unlike [32], which employs adversarial optimization, our method is entirely training-free. Additionally, we provide both empirical and theoretical validation for the choice of the intermediate time step. To the best of our knowledge, this is the first training-free method that supports watermark embedding for both server-side and user-side applications while maintaining high robustness and consistency.

B.2 Low-dimensional Subspace in Diffusion Model

In recent years, there has been growing interest in understanding deep generative models through the lens of the manifold hypothesis [70]. This hypothesis suggests that high-dimensional real-world data actually lies in latent manifolds with a low intrinsic dimension. Focusing on diffusion models, [71] empirically and theoretically shows that the approximated score function (the gradient of the log density of a noise-corrupted data distribution) in diffusion models is orthogonal to a low-dimensional subspace. Building on this, [33, 34] find that the estimated posterior mean from diffusion models lies within this low-dimensional space. Additionally, [34] discovers strong local linearity within the space, suggesting that it can be locally approximated by a linear subspace. This observation motivates our Assumption 1, where we assume the estimated posterior mean lies in a low-dimensional subspace.

Building upon these findings, [71, 72] introduce a local intrinsic dimension estimator, while [70] proposes a method for detecting out-of-domain data. [33] offers theoretical insights into how diffusion model training transitions from memorization to generalization, and [34, 41] explores the semantic basis of the subspace to achieve disentangled image editing. Unlike these previous works, our approach leverages the low-dimensional subspace for watermarking, where both empirical and theoretical evidence demonstrates that this subspace enhances robustness and consistency.

C Additional Experiments

C.1 Details about Attacks

In this work, we intensively tested our method on four different watermarking attacks, both in the server scenario and in the user scenario. These watermarking attacks can be categorized into three groups, including:

- **Distortion attack**
  - JPEG compression (JPEG) with a compression rate of 25%.
  - Gaussian blurring (G.Blur) with an 8×8 filter size.
  - Gaussian noise (G.Noise) with $\sigma = 0.1$.
  - Color jitter (CJ) with brightness factor uniformly ranges between 0 and 6.
  - Resize and restore (RR). Resize to 50% of pixels and restore to original size.
  - Random drop (RD). Random drop a square with 40% of pixels.
  - Median blurring (M.Blur) with a 7×7 median filter.
- **Regeneration attack**
  - Diffusion purification [60] (DiffPure) with the purified step at 0.3T.
  - VAE-based image compression [61] (IC1) and [62] (IC2), with a quality level of 3.
  - Diffusion-based image regeneration (IR) [63].
  - Rinsing regenerations [59]) with 2 times (Rinse2x) and 4 times (Rinse4x).
- **Adversarial attack**
  - Blackbox averaging (BA) and greybox averaging (GA) watermarking removal attack [64].

Visualizations of these attacks are in Figure 4. Detailed experiments for Table 1 (Table 2) on the above attacks are reported by groups, with the distortion attack in Table 3 (Table 5) and the regeneration and adversarial attacks in Table 4 (Table 6)

(a) **Clean** (b) **JPEG** (c) **G.Blur** (d) **G.Noise** (e) **CJ** (f) **RR** (g) **RD**

(h) **M.Blur** (i) **DiffPure** (j) **IC1** (k) **IC2** (l) **IR** (m) **Rinse2x** (n) **Rinse4x**

Figure 4: **Visualization of different attacks.**

C.2 Multi-key Watermarking

In this section, we examine the capability of Shallow Diffuse to support multi-key watermarking. We evaluate the ability to embed multiple watermarks into the same image and detect each one independently. For this experiment, we test cases with 2, 4, 8, 16, 32 watermarks. Each watermark uses a unique ring-shaped key $\boldsymbol{W}_i$ and a non-overlapped mask M (part of a circle). This is a non-trivial setting as we could pre-defined the key number and non-overlapped mask M for application.

Table 3: **Watermarking Robustness for distortion attacks under the server scenario.**

Method	Watermarking Robustness (AUC ↑/TPR@1%FPR↑)								
	Clean	JPEG	G.Blur	G.Noise	CJ	RR	RD	M.Blur	Distortion Average
DwtDct	0.97/0.85	0.47/0.00	0.51/0.02	0.96/0.78	0.53/0.15	0.66/0.14	0.99/0.88	0.58/0.01	0.71/0.35
DwtDctSvd	**1.00/1.00**	0.64/0.10	0.96/0.70	0.99/0.99	0.53/0.12	0.99/0.99	**1.00/1.00**	**1.00/1.00**	0.89/0.74
RivaGAN	1.00/0.99	0.94/0.69	0.96/0.76	0.97/0.88	0.95/0.79	0.99/0.98	0.99/0.98	0.99/0.97	0.97/0.88
Stegastamp	**1.00/1.00**	**1.00/1.00**	1.00/0.95	0.98/0.97	1.00/0.97	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99
Stable Signature	**1.00/1.00**	0.99/0.76	0.57/0.00	0.71/0.14	0.96/0.87	0.90/0.34	**1.00/1.00**	0.95/0.62	0.89/0.59
Tree-Ring Watermarks	**1.00/1.00**	0.99/0.97	0.98/0.98	0.94/0.50	0.96/0.67	**1.00/1.00**	0.99/0.97	0.99/0.94	0.98/0.88
RingID	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99	0.99/0.98	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**
Gaussian Shading	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**
Shallow Diffuse (ours)	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**

Table 4: **Watermarking Robustness for regeneration and adversarial attacks under the server scenario.**

Method	Watermarking Robustness (AUC ↑/TPR@1%FPR↑)									
	DiffPure	IC1	IC2	IR	Rinse2x	Rinse4x	Regeneration Average	BA	GA	Adversarial Average
DwtDct	0.50/0.00	0.52/0.01	0.49/0.00	0.50/0.00	0.77/0.04	0.80/0.03	0.60/0.01	0.27/0.00	0.99/0.84	0.63/0.42
DwtDctSvd	0.51/0.02	0.73/0.03	0.68/0.04	0.70/0.07	0.78/0.18	0.78/0.10	0.70/0.07	0.86/0.02	0.17/0.00	0.52/0.01
RivaGAN	0.73/0.16	0.65/0.03	0.63/0.04	0.56/0.00	0.64/0.03	0.58/0.02	0.63/0.05	0.94/0.64	1.00/1.00	0.97/0.82
Stegastamp	0.81/0.29	1.00/0.97	1.00/0.99	0.90/0.43	0.75/0.13	0.67/0.06	0.85/0.48	0.63/0.03	0.68/0.06	0.66/0.05
Stable Signature	0.54/0.01	0.93/0.58	0.91/0.50	0.67/0.02	0.64/0.01	0.54/0.01	0.71/0.19	1.00/0.98	1.00/1.00	**1.00/0.99**
Tree-Ring Watermarks	0.98/0.73	0.99/0.97	0.99/0.98	0.99/0.92	0.98/0.88	0.96/0.75	0.98/0.87	0.16/0.08	0.05/0.03	0.11/0.06
RingID	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99	**1.00/1.00**	0.44/0.35	0.40/0.31	0.42/0.33
Gaussian Shading	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.53/0.48	0.52/0.47	0.53/0.47
Shallow Diffuse (ours)	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99	1.00/1.00	0.99/0.90	1.00/0.98	0.57/0.45	0.70/0.63	0.64/0.54

Table 5: **Watermarking Robustness for distortion attacks under the user scenario.**

Method	Watermarking Robustness (AUC ↑/TPR@1%FPR↑)								
	Clean	JPEG	G.Blur	G.Noise	CJ	RR	RD	M.Blur	Average
COCO Dataset									
DwtDct	0.98/0.83	0.50/0.01	0.50/0.00	0.97/0.81	0.54/0.14	0.67/0.17	0.99/0.93	0.59/0.05	0.64/0.54
DwtDctSvd	**1.00/1.00**	0.64/0.13	0.98/0.83	0.99/0.99	0.54/0.13	1.00/1.00	1.00/1.00	**1.00/1.00**	0.89/0.76
RivaGAN	**1.00/1.00**	0.97/0.86	0.98/0.86	0.99/0.94	0.96/0.82	1.00/1.00	1.00/1.00	**1.00/1.00**	0.99/0.93
Stegastamp	**1.00/1.00**	**1.00/1.00**	0.99/0.90	0.90/0.87	1.00/0.98	1.00/0.99	1.00/0.99	**1.00/1.00**	0.99/0.97
Tree-Ring Watermarks	**1.00/1.00**	0.99/0.87	0.99/0.86	**1.00/1.00**	0.88/0.49	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.98/0.90
RingID	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.98/0.86	**1.00/0.99**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.98
Gaussian Shading	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.95	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99
Shallow Diffuse (ours)	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/0.99**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**
DiffusionDB Dataset									
DwtDct	0.96/0.76	0.47/0.002	0.51/0.018	0.96/0.78	0.53/0.15	0.66/0.14	0.99/0.88	0.58/0.01	0.71/0.34
DwtDctSvd	**1.00/1.00**	0.64/0.10	0.96/0.70	0.99/0.99	0.53/0.12	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.89/0.74
RivaGAN	1.00/0.98	0.94/0.69	0.96/0.76	0.97/0.88	0.95/0.79	1.00/0.98	0.99/0.98	**1.00/1.00**	0.98/0.88
Stegastamp	**1.00/1.00**	**1.00/1.00**	0.99/0.88	0.91/0.89	1.00/0.99	1.00/0.97	**1.00/1.00**	1.00/0.96	0.99/0.96
Tree-Ring Watermarks	**1.00/1.00**	0.99/0.68	0.94/0.62	**1.00/1.00**	0.84/0.15	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.97/0.81
RingID	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.98/0.86	1.00/0.98	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.98
Gaussian Shading	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.99/0.96	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99
Shallow Diffuse (ours)	**1.00/1.00**	1.00/0.99	1.00/0.99	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**

Table 6: **Watermarking Robustness for regeneration and adversarial attacks under the user scenario.**

Method	Watermarking Robustness (AUC ↑/TPR@1%FPR↑)									
	DiffPure	IC1	IC2	IR	Rinse2x	Rinse4x	Regeneration Average	BA	GA	Adversarial Average
COCO Dataset										
DwtDct	0.46/0.00	0.49/0.00	0.49/0.01	0.46/0.00	0.61/0.00	0.65/0.01	0.53/0.00	0.97/0.80	0.96/0.84	0.97/0.82
DwtDctSvd	0.50/0.01	0.70/0.05	0.64/0.04	0.68/0.07	0.72/0.08	0.69/0.08	0.66/0.06	0.79/0.00	0.49/0.00	0.64/0.00
RivaGAN	0.63/0.02	0.68/0.05	0.66/0.04	0.75/0.15	0.75/0.04	0.68/0.03	0.69/0.05	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**
Stegastamp	0.81/0.27	1.00/0.95	1.00/0.95	0.85/0.28	0.78/0.23	0.69/0.16	0.86/0.47	0.73/0.23	0.71/0.28	0.72/0.26
Tree-Ring Watermarks	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.99/0.92	0.98/0.78	1.00/0.95	0.60/0.39	0.46/0.23	0.53/0.31
RingID	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.75/0.59	**1.00/1.00**	0.88/0.79
Gaussian Shading	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.53/0.48	0.52/0.47	0.53/0.47
Shallow Diffuse (ours)	0.99/0.86	1.00/0.99	0.99/0.97	**1.00/1.00**	**1.00/1.00**	1.00/0.93	1.00/0.96	0.70/0.62	0.70/0.62	0.70/0.62
DiffusionDB Dataset										
DwtDct	0.50/0.00	0.52/0.01	0.49/0.00	0.50/0.00	0.64/0.00	0.66/0.02	0.55/0.01	0.97/0.79	0.97/0.77	0.97/0.78
DwtDctSvd	0.51/0.02	0.73/0.03	0.68/0.04	0.70/0.07	0.73/0.07	0.66/0.02	0.67/0.04	0.77/0.00	0.39/0.00	0.58/0.00
RivaGAN	0.56/0.00	0.65/0.03	0.63/0.04	0.73/0.16	0.70/0.02	0.63/0.01	0.65/0.04	1.00/0.98	1.00/0.99	**1.00/0.98**
Stegastamp	0.83/0.28	1.00/0.91	1.00/0.93	0.85/0.40	0.78/0.13	0.68/0.11	0.86/0.46	0.69/0.21	0.71/0.30	0.70/0.26
Tree-Ring Watermarks	0.99/0.99	0.99/0.99	0.99/0.98	0.96/0.92	0.98/0.81	0.95/0.54	0.98/0.87	0.51/0.32	0.38/0.20	0.45/0.26
RingID	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99	**1.00/1.00**	0.71/0.58	**1.00/1.00**	0.85/0.79
Gaussian Shading	1.00/0.99	1.00/0.99	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.50/0.46	0.50/0.46	0.50/0.46
Shallow Diffuse (ours)	0.96/0.90	0.96/0.92	0.97/0.93	0.98/0.96	1.00/0.98	0.98/0.88	0.97/0.93	0.66/0.58	0.68/0.60	0.67/0.59

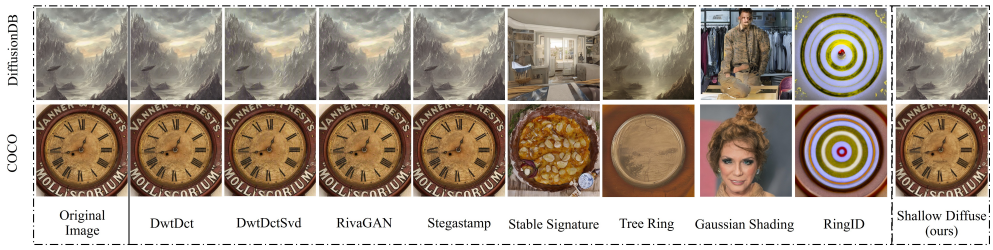


Figure 5: **Generation Consistency in User Scenarios.** We compare the visualization quality of our method against DwtDct, DwtdctSvd, RivaGAN, Stegastamp, Tree Ring, Gaussian Shading, and RingID across the DiffusionDB, and COCO datasets.

Figure 6: **Generation Consistency in server scenarios.** We compare the visualization quality of our method against the original image and StageStamp.

The metric for this task is the average robustness across all keys, measured in terms of AUC and TPR@1%FPR. For this study, we test the Tree-Ring and Shallow Diffuse in the server scenario. The results of this experiment are presented in Table 7. Shallow Diffuse consistently outperformed Tree-Ring in robustness across different numbers of users. Even as the number of users increased to 32, Shallow Diffuse maintained strong robustness under clean conditions. However, in adversarial settings, its robustness began to decline when the number of users exceeded 16. Under the current setup, when the number of users surpasses the predefined limit, our method becomes less robust and accurate. We believe that enabling watermarking for hundreds or even thousands of users simultaneously is a challenging yet promising future direction for Shallow Diffuse.

Table 7: **Multi-key re-watermark for different attacks under the server scenario.**

Watermark numbder	Method	Watermarking Robustness (AUC ↑/TPR@1%FPR↑)												
		Clean	JPEG	G.Blur	G.Noise	CJ	RR	RD	M.Blur	DiffPure	IC1	IC2	IR	Average
2	Tree-Ring	**1.00/1.00**	0.99/0.84	1.00/0.97	0.95/0.83	0.98/0.75	1.00/1.00	1.00/1.00	1.00/1.00	0.91/0.23	1.00/0.91	0.98/0.82	0.94/0.49	0.98/0.80
	Shallow Diffuse	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**0.98/0.95**	**1.00/0.90**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**0.98/0.65**	**1.00/0.91**	**1.00/0.97**	**1.00/0.99**	**0.99/0.95**
4	Tree-Ring	**1.00/1.00**	0.98/0.63	1.00/0.89	0.96/0.86	0.90/0.54	1.00/0.92	1.00/0.99	1.00/0.95	0.88/0.11	0.99/0.72	0.97/0.67	0.92/0.37	0.96/0.70
	Shallow Diffuse	**1.00/1.00**	1.00/0.96	0.99/0.88	0.97/0.91	0.99/0.82	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.94/0.37	0.99/0.80	0.99/0.83	0.99/0.89	0.99/0.86
8	Tree-Ring	1.00/0.95	0.90/0.32	0.97/0.56	0.92/0.64	0.90/0.45	0.98/0.71	1.00/0.89	0.98/0.68	0.77/0.08	0.91/0.38	0.89/0.25	0.83/0.16	0.91/0.47
	Shallow Diffuse	**1.00/1.00**	0.99/0.85	0.97/0.73	0.97/0.90	0.98/0.80	1.00/0.98	**1.00/1.00**	1.00/0.96	0.91/0.36	0.98/0.71	0.97/0.70	0.99/0.80	0.98/0.80
16	Tree-Ring	0.96/0.57	0.78/0.18	0.87/0.32	0.87/0.38	0.84/0.24	0.90/0.42	0.95/0.53	0.90/0.36	0.68/0.05	0.80/0.18	0.77/0.14	0.72/0.05	0.83/0.26
	Shallow Diffuse	1.00/0.89	0.94/0.59	0.89/0.39	0.94/0.73	0.92/0.53	0.97/0.73	0.99/0.84	0.96/0.73	0.78/0.11	0.90/0.46	0.91/0.46	0.92/0.55	0.92/0.56
32	Tree-Ring	0.95/0.44	0.77/0.11	0.85/0.15	0.86/0.31	0.80/0.15	0.88/0.22	0.94/0.34	0.89/0.26	0.63/0.03	0.78/0.11	0.75/0.08	0.70/0.05	0.80/0.16
	Shallow Diffuse	0.99/0.89	0.91/0.46	0.86/0.26	0.93/0.63	0.91/0.47	0.96/0.65	0.99/0.84	0.95/0.59	0.74/0.07	0.87/0.31	0.87/0.30	0.89/0.28	0.90/0.44

C.3 Ablation Study of Different Watermark Patterns

In Table 8, we examine various combinations of watermark patterns $\boldsymbol{M} \odot \boldsymbol{W}$. For the shape of the mask $\boldsymbol{M}$, "Circle" refers to a circular mask $\boldsymbol{M}$ (see Figure 3 top left), while "Ring" represents a

ring-shaped $\boldsymbol{M}$. Since the mask is centered in the middle of the figure, "Low" and "High" denote frequency regions: "Low" represents a DFT with zero-frequency centering, whereas "High" indicates a DFT without zero-frequency centering, as illustrated in Figure 3 bottom. For the distribution of $\boldsymbol{W}$, "Zero" implies all values are zero, "Rand" denotes values sampled from $\mathcal{N}(\mathbf{0}, \mathbf{1})$, and "Rotational Rand" represents multiple concentric rings in $\boldsymbol{W}$, with each ring's values sampled from $\mathcal{N}(\mathbf{0}, \mathbf{1})$.

As shown in Table 8, watermarking in high-frequency regions (Rows 7-9) yields improved image consistency compared to low-frequency regions (Rows 1-6). Additionally, the "Circle" $\boldsymbol{M}$ combined with "Rotational Rand" $\boldsymbol{W}$ (Rows 3 and 9) demonstrates greater robustness than other watermark patterns. Consequently, Shallow Diffuse employs the "Circle" $\boldsymbol{M}$ with "Rotational Rand" $\boldsymbol{W}$ in the high-frequency region.

Table 8: **Ablation study on different watermark patterns.**

Method & Dataset			PSNR ↑	SSIM ↑	LPIPS ↓	Average Watermarking Robustness (AUC ↑/TPR@1%FPR↑)
Frequency Region	Shape	Distribution				
Low	Circle	Zero	29.10	0.90	0.06	0.93/0.65
Low	Circle	Rand	29.37	0.92	0.05	0.92/0.25
Low	Circle	Rotational Rand	29.13	0.90	0.06	**1.00/1.00**
Low	Ring	Zero	36.20	0.95	0.02	0.78/0.35
Low	Ring	Rand	38.23	0.97	0.01	0.87/0.49
Low	Ring	Rotational Rand	35.23	0.93	0.02	0.99/0.98
High	Circle	Zero	38.3	0.96	0.01	0.80/0.34
High	Circle	Rand	**42.3**	**0.98**	**0.004**	0.86/0.35
High	Circle	Rotational Rand	38.0	0.94	0.01	**1.00/1.00**

C.4 Ablation Study of Watermarking Embedded Channel.

As shown in Table 9, we evaluate specific embedding channels c for Shallow Diffuse, where "0," "1," "2," and "3" denote $c = 0, 1, 2, 3$, respectively, and "0 + 1 + 2 + 3" indicates watermarking applied across all channels [3]. Since applying watermarking to any single channel yields similar results (Row 1-4), but applying it to all channels (Row 5) negatively impacts image consistency and robustness, we set $c = 3$ for Shallow Diffuse. The reason is that many image processing operations tend to affect all channels uniformly, making watermarking across all channels more susceptible to such attacks.)

Table 9: **Ablation study on watermarking embedded channel.**

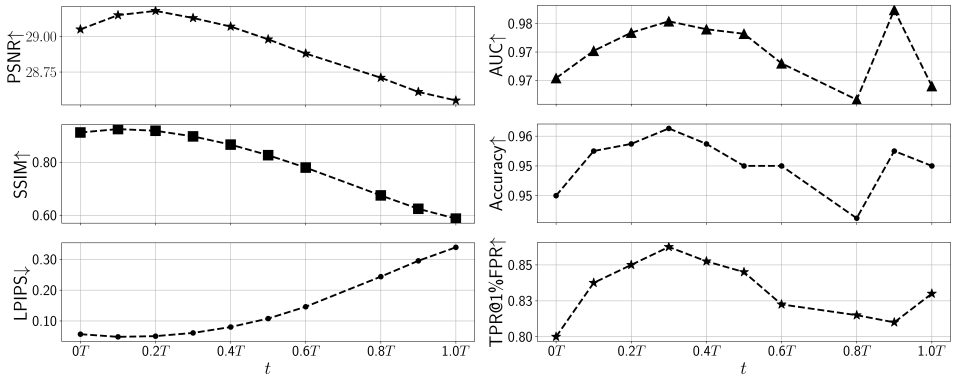
Watermark embedding channel	PSNR ↑	SSIM ↑	LPIPS ↓	Watermarking Robustness (TPR@1%FPR↑)				
				Clean	JPEG	G.Blur	G.Noise	Color Jitter
0	36.46	**0.93**	**0.02**	**1.00**	**1.00**	**1.00**	**1.00**	0.99
1	36.57	**0.93**	**0.02**	**1.00**	**1.00**	**1.00**	**1.00**	0.99
2	36.13	0.92	**0.02**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
3	**36.64**	**0.93**	**0.02**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
0 + 1 + 2 + 3	33.19	0.83	0.05	**1.00**	**1.00**	**1.00**	**1.00**	0.95

C.5 Ablation Study over Injecting Timesteps.

Figure 7 shows the relationship between the watermark injection timestep t and both consistency and robustness [4]. Shallow Diffuse achieves optimal consistency at $t = 0.2T$ and optimal robustness at $t = 0.3T$. In practice, we select $t = 0.3T$. This result aligns with the intuitive idea proposed in Section 3.1 and the theoretical analysis in Section 4: low-dimensionality enhances both data generation consistency and watermark detection robustness. However, according to [34], the optimal timestep r_t for minimizing r_t satisfies $t^* \in [0.5T, 0.7T]$. We believe the best consistency and robustness are not achieved at t^* due to the error introduced by $\texttt{DDIM} - \texttt{Inv}$. As t increases, this error grows, leading to a decline in both consistency and robustness. Therefore, the best tradeoff is reached at $t \in [0.2T, 0.3T]$, where $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ remains low-rank but t is still below t^*. Another possible explanation is the gap between the image space and latent space in diffusion models. The

[3] Here we apply Shallow Diffuse on the latent space of Stable Diffusion, the channel dimension is 4.

[4] In this experiment, we do not incorporate additional techniques like channel averaging or enhanced watermark patterns. Therefore, when $t = 1.0T$, the method is equivalent to Tree-Ring.



(a) **Consistency** (b) **Robustness**

Figure 7: **Ablation study at different timestep t.**

rank curve in [34] is evaluated for an image-space diffusion model, whereas Shallow Diffuse operates in the latent-space diffusion model (e.g., Stable Diffusion).

C.6 Ablation Study of Inference Steps

We conducted ablation studies on the number of sampling steps, across 10, 25, and 50 steps. The results, shown in Table 10, indicate that Shallow Diffuse is not highly sensitive to sampling steps. The watermark robustness remains consistent across all sampling steps.

Table 10: **Ablation study over inference steps.**

Steps	Watermarking Robustness (AUC ↑/TPR@1%FPR↑)													
	Clean	G.Noise	CJ	RD	M.Blur	DiffPure	IC1	IC2	DiffDeeper	Rinse2x	Rinse4x	BA	GA	Average
10	**1.00/1.00**	0.99/0.89	0.95/0.76	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.99/0.93	0.99/0.93	1.00/0.99	**1.00/1.00**	**1.00/0.98**	**0.63/0.49**	**0.74/0.70**	0.95/0.90
25	**1.00/1.00**	1.00/0.97	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.99/0.91	0.99/0.92	**1.00/1.00**	**1.00/1.00**	1.00/0.92	0.56/0.48	0.73/0.65	0.94/0.91
50	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99	1.00/1.00	0.99/0.90	0.57/0.45	0.70/0.63	**0.94/0.92**

D Robustness Analysis on Geometric Distortions

To further analyze robustness under geometric transformations, we conducted an extended study focusing on rotation, cropping, and scaling.

Controllable Trade-off via Mask Radius. Our framework enables explicit control over the balance between perceptual quality and geometric robustness by adjusting the frequency mask radius r. We compared the original configuration (optimized for visual fidelity) to a more robust configuration with an expanded radius ($r = 3$–13). Results in Table 11 show that increasing the radius improves geometric robustness—particularly rotation and cropping—while incurring only a mild degradation in image consistency.

Table 11: **Trade-off between image fidelity and geometric robustness.** Increasing the mask radius ($r = 3$–13) enhances rotation and cropping robustness with minor PSNR drop.

Setting	CLIP	PSNR ↑	SSIM ↑	LPIPS ↓	Watermarking Robustness (AUC ↑)		
					Rotation	Cropping	Scaling
Current ($r = 0$–10)	0.3669	35.49	0.96	0.02	0.68	0.56	1.00
Robust ($r = 3$–13)	0.3637	32.05	0.95	0.03	0.90	0.89	1.00

E Generalization to Transformer-Based Diffusion Models

To assess whether Shallow Diffuse generalizes beyond U-Net based diffusion architectures, we conducted an additional study on FLUX [73], a transformer-based diffusion model that employs a Flow Matching noise scheduler.

Experimental Setup. All evaluations were performed under a server-side watermarking scenario at 512×512 resolution. The same watermark design as in the Stable Diffusion experiments was used, with two key modifications to account for architectural differences: (a) the watermark radius was set to 5, and (b) watermark injection was applied across all latent channels.

We generated 100 watermarked images and evaluated both consistency and robustness across injection timesteps $\{0.1T, 0.2T, \ldots, 0.9T\}$. The results are presented in Table 12.

Table 12: **Generalization of Shallow Diffuse to the FLUX transformer-based diffusion model.**

Timestep (t/T)	PSNR ↑	SSIM ↑	LPIPS ↓	Watermarking Robustness		
				AUC ↑	ACC ↑	TPR@1%FPR ↑
0.1	31.27	0.91	0.05	0.90	0.93	**0.87**
0.2	31.66	0.92	**0.04**	0.91	0.93	**0.87**
0.3	31.68	0.92	0.05	0.92	**0.94**	**0.87**
0.4	**31.69**	**0.93**	**0.04**	0.93	**0.94**	0.86
0.5	31.68	**0.93**	**0.04**	**0.94**	0.93	0.86
0.6	31.56	**0.93**	**0.04**	**0.94**	**0.94**	0.85
0.7	31.50	**0.93**	**0.04**	**0.94**	0.93	0.81
0.8	31.52	**0.93**	0.05	**0.94**	**0.94**	0.82
0.9	31.62	0.92	0.06	**0.94**	**0.94**	0.81

Analysis. To ensure a fair comparison of injection timesteps across different schedulers, we matched the effective Signal-to-Noise Ratio (SNR) between the Variance Preserving (VP) schedule used in Stable Diffusion and the Flow Matching scheduler in FLUX. A timestep of $t/T = 0.3$ in VP approximately corresponds to $t/T = 0.205$ in Flow Matching when equalizing SNR.

The results indicate that injecting at this equivalent "shallow" timestep achieves the best SSIM (0.93) and near-optimal PSNR (31.7), while also maximizing robustness (AUC 0.94, TPR@1%FPR 0.87). This confirms that the optimal embedding region discovered for Stable Diffusion generalizes to transformer-based architectures, underscoring the broad applicability of our null-space embedding framework.

F Comparison with ROBIN

We conduct a direct empirical comparison between our optimization-free Shallow Diffuse and the optimization-based ROBIN [32] under the server scenario with 1000 generations. Experiment results are shown in Table 13 and Table 14.

Our experiments show that Shallow Diffuse produces images with significantly higher perceptual quality and consistency. As shown in Table 13, our method achieves a PSNR nearly 11 dB higher and a better FID score. We attribute the difference to the frequency domains where watermarks are added: Shallow Diffuse uses the high-frequency domain, while ROBIN uses the low-frequency domain. Adding the watermark to high frequencies preserves the low-frequency content of the generated image, thereby significantly improving consistency and quality.

In terms of robustness, the two methods are competitive, each with distinct strengths. Table 14 shows that both methods are highly robust to most distortion and regeneration attacks. ROBIN demonstrates superior robustness against geometric attacks like rotation (1.0 vs 0.69) and cropping (0.99 vs 0.58), as well as adversarial attacks.

This empirical comparison quantifies the fundamental trade-off. ROBIN's optimization process achieves higher robustness for challenging attacks at the cost of significantly lower image quality, longer setup times, and less flexibility. Shallow Diffuse provides a more balanced and practical

solution, offering state-of-the-art image quality and comparable robustness across a wide range of common attacks, all within an efficient, optimization-free framework adaptable to both server and user needs.

Table 13: **Consistency between Shallow Diffuse and ROBIN under the server scenario.**

Method	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	CLIP ↑
ROBIN	24.614	0.8261	0.1087	134.8	0.366
Shallow Diffuse	**35.49**	**0.96**	**0.05**	**129.228**	**0.367**

Table 14: **Robustness between Shallow Diffuse and ROBIN under the server scenario.**

Method	Watermarking Robustness (AUC) ↑														
	JPEG	G.Blur	G.Noise	CJ	RR	RD	M.Blur	Rotation	Crop	IC1	IC2	IR	Rinse4x	BA	GA
ROBIN	**0.999**	**0.999**	0.963	0.962	**1**	**1**	**1**	**1**	**0.991**	0.998	**1**	**1**	**1**	**0.939**	**0.9**
Shallow Diffuse	**0.999**	**0.999**	**0.997**	**0.967**	**1**	**1**	**1**	0.691	0.582	**1**	0.998	0.993	0.999	0.67	0.78

G Proofs in Section 4

G.1 Proofs of Theorem 1

Proof of Theorem 1. According to Assumption 1, we have $||\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t}||_2^2 = \lambda||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \cdot \Delta\boldsymbol{x}||_2^2$. From Levy's Lemma proposed in [74], given function $||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \cdot \Delta\boldsymbol{x}||_2^2 : \mathbb{S}^{d-1} \to \mathbb{R}$ we have:

$$\mathbb{P}\left(\left|||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \cdot \Delta\boldsymbol{x}||_2^2 - \mathbb{E}\left[||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \cdot \Delta\boldsymbol{x}||_2^2\right]\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-C(d-2)\epsilon^2}{L^2}\right),$$

given L to be the Lipschitz constant of $||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_2^2$ and C is a positive constant (which can be taken to be $C = (18\pi^3)^{-1}$). From Lemma 2 and Lemma 3, we have:

$$\mathbb{P}\left(\left|||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \cdot \Delta\boldsymbol{x}||_2^2 - \frac{||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_F^2}{d}\right| \geq \epsilon\right) \leq 2\exp\left(\frac{-(18\pi^3)^{-1}(d-2)\epsilon^2}{||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_2^4}\right).$$

Define $\dfrac{1}{r_t}$ as the desired probability level, set

$$\frac{1}{r_t} = 2\exp\left(\frac{-(18\pi^3)^{-1}(d-2)\epsilon^2}{||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_2^4}\right),$$

Solving for ϵ:

$$\epsilon = ||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_2^2\sqrt{\frac{18\pi^3}{d-2}\log(2r_t)}.$$

Therefore, with probability $1 - \dfrac{1}{r_t}$, we have:

$$\begin{aligned}
||\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t}||_2^2 &= \lambda^2||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \cdot \Delta\boldsymbol{x}||_2^2, \\
&\leq \frac{\lambda^2||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_F^2}{d} + \lambda^2||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_2^2\sqrt{\frac{18\pi^3}{d-2}\log(2r_t)}, \\
&\leq \lambda^2||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_2^2\left(\frac{r_t}{d} + \sqrt{\frac{18\pi^3}{d-2}\log(2r_t)}\right), \\
&= \lambda^2 L^2\left(\frac{r_t}{d} + \sqrt{\frac{18\pi^3}{d-2}\log(2r_t)}\right),
\end{aligned}$$

where the last inequality is obtained from $||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_F^2 \leq r_t||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)||_2^2$. Therefore, with probability $1 - \frac{1}{r_t}$,

$$||\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t}||_2 \leq \lambda L \sqrt{\frac{r_t}{d} + \sqrt{\frac{18\pi^3}{d-2}\log(2r_t)}} = \lambda L h(r_t).$$

□

Proof of Theorem 2. According to Equation (1), one step of DDIM sampling at timestep t could be represented by PMP $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ as:

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \sqrt{1-\alpha_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)}{\sqrt{1-\alpha_t}}\right), \tag{10}$$

$$= \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}}\boldsymbol{x}_t + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t), \tag{11}$$

If we inject a watermark $\lambda\Delta\boldsymbol{x}$ to $\boldsymbol{x}_t$, so $x_t^{\mathcal{W}} = \boldsymbol{x}_t + \lambda\Delta\boldsymbol{x}$. To solve $x_{t-1}^{\mathcal{W}}$, we could plugging Equation (2) to Equation (11), we could obtain:

$$\boldsymbol{x}_{t-1}^{\mathcal{W}} = \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}}\boldsymbol{x}_t^{\mathcal{W}} + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t^{\mathcal{W}}), \tag{12}$$

$$= \boldsymbol{x}_{t-1} + \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}}\lambda\Delta\boldsymbol{x} + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}}\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x} \tag{13}$$

$$= \boldsymbol{x}_{t-1} + \lambda \underbrace{\left(\sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}}\boldsymbol{I} + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}}\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\right)}_{:=\boldsymbol{W}_t}\Delta\boldsymbol{x}, \tag{14}$$

One step DDIM Inverse sampling at timestep $t-1$ could be represented by PMP $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ as:

$$\boldsymbol{x}_t = \sqrt{\frac{1-\alpha_t}{1-\alpha_{t-1}}}\boldsymbol{x}_{t-1} + \frac{\sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t} - \sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}}}{\sqrt{1-\alpha_{t-1}}}\boldsymbol{f}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1}), \tag{15}$$

To detect the watermark, we apply one step DDIM Inverse on $\boldsymbol{x}_{t-1}^{\mathcal{W}}$ at timestep $t-1$ to obtain $\tilde{x}_t^{\mathcal{W}}$:

$$\tilde{\boldsymbol{x}}_t^{\mathcal{W}} = \sqrt{\frac{1-\alpha_t}{1-\alpha_{t-1}}}\boldsymbol{x}_{t-1}^{\mathcal{W}} + \frac{\sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t} - \sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}}}{\sqrt{1-\alpha_{t-1}}}\boldsymbol{f}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1}^{\mathcal{W}}),$$

$$= \boldsymbol{x}_t + \lambda \underbrace{\left(\sqrt{\frac{1-\alpha_t}{1-\alpha_{t-1}}}\boldsymbol{I} + \frac{\sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t} - \sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}}}{\sqrt{1-\alpha_{t-1}}}\boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1})\right)}_{:=\boldsymbol{W}_{t-1}}\boldsymbol{W}_t\Delta\boldsymbol{x},$$

$$= \boldsymbol{x}_t + \lambda\boldsymbol{W}_{t-1}\boldsymbol{W}_t\Delta\boldsymbol{x} = \boldsymbol{x}_t^{\mathcal{W}} + \lambda\left(\boldsymbol{W}_{t-1}\boldsymbol{W}_t - \boldsymbol{I}\right)\Delta\boldsymbol{x}.$$

Therefore:

$$
\begin{aligned}
||\tilde{x}_t^{\mathcal{W}} - \boldsymbol{x}_t^{\mathcal{W}}||_2 &= \lambda || \left(\boldsymbol{W}_{t-1}\boldsymbol{W}_t - \boldsymbol{I}\right) \Delta\boldsymbol{x}||_2, \\
&= \lambda || \frac{\sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t} - \sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}}}{\sqrt{1-\alpha_t}} \boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1})\Delta\boldsymbol{x}, \\
&\quad + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_{t-1}}} \boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}, \\
&\quad - \frac{\left(\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}\right)^2}{\sqrt{1-\alpha_{t-1}}\sqrt{1-\alpha_t}} \boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1})\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}||_2, \\
&\leq -\lambda g\left(\alpha_t, \alpha_{t-1}\right) ||\boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1})\Delta\boldsymbol{x}||_2 + \lambda g\left(\alpha_{t-1}, \alpha_t\right) ||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}||_2 \\
&\quad - \lambda g\left(\alpha_{t-1}, \alpha_t\right) g\left(\alpha_t, \alpha_{t-1}\right) ||\boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1})\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}||_2, \\
&\leq -\lambda g\left(\alpha_t, \alpha_{t-1}\right) ||\boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1})\Delta\boldsymbol{x}||_2 \\
&\quad + \lambda g\left(\alpha_{t-1}, \alpha_t\right) \left(1 - g\left(\alpha_t, \alpha_{t-1}\right) L\right) ||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}||_2, \\
&= -g\left(\alpha_t, \alpha_{t-1}\right) ||\hat{\boldsymbol{x}}_{0,t-1}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t-1}||_2 \\
&\quad + g\left(\alpha_{t-1}, \alpha_t\right) \left(1 - g\left(\alpha_t, \alpha_{t-1}\right) L\right) ||\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t}||_2,
\end{aligned}
$$

The first inequality holds because $g\left(\alpha_{t-1}, \alpha_t\right) < 0$ and $g\left(\alpha_t, \alpha_{t-1}\right) > 0$. The second inequality holds because $||\boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1})\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}||_2 \leq ||\boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1})||_2 ||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}||_2 \leq L||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\Delta\boldsymbol{x}||_2$. From Theorem 1, with probability $1 - \frac{1}{r_{t-1}}$,

$$
||\hat{\boldsymbol{x}}_{0,t-1}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t-1}||_2 \leq \lambda L h(r_{t-1}),
$$

with probability $1 - \frac{1}{r_t}$,

$$
||\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t}||_2 \leq \lambda L h(r_t),
$$

Thus, from the union of bound, with a probability at least $1 - \frac{1}{r_t} - \frac{1}{r_{t-1}}$,

$$
\begin{aligned}
||\tilde{x}_t^{\mathcal{W}} - \boldsymbol{x}_t^{\mathcal{W}}||_2 &\leq -\lambda L g\left(\alpha_t, \alpha_{t-1}\right) h(r_{t-1}) + \lambda L g\left(\alpha_{t-1}, \alpha_t\right) \left(1 - g\left(\alpha_t, \alpha_{t-1}\right) L\right) h(r_t) \\
&\leq \lambda L \left(-g\left(\alpha_t, \alpha_{t-1}\right) + g\left(\alpha_{t-1}, \alpha_t\right) \left(1 - L g\left(\alpha_t, \alpha_{t-1}\right)\right)\right) h(\max\{r_{t-1}, r_t\})
\end{aligned}
$$

□

H Auxiliary Results

Lemma 1. *Given a unit vector $\boldsymbol{v}_i$ with and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, we have*

$$
\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}[\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2 / ||\boldsymbol{\epsilon}||_2^2] = \frac{1}{d}.
$$

Proof of Lemma 1. Because $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$,

$$
\boldsymbol{v}_i^T\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{v}_i^T\boldsymbol{0}, \boldsymbol{v}_i^T\boldsymbol{I}_d\boldsymbol{v}_i) = \mathcal{N}(\boldsymbol{v}_i^T\boldsymbol{0}, \boldsymbol{v}_i^T\boldsymbol{I}_d\boldsymbol{v}_i) = \mathcal{N}(0, 1), \tag{16}
$$

Assume a set of d unit vecotrs $\{v_1, v_2, \ldots, \boldsymbol{v}_i, \ldots, v_d\}$ are orthogonormal and are basis of $\mathbb{R}^d$, similarly, we could show that $\forall j \in [d], X_j \coloneqq v_j^T\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$. Therefore, we could rewrite $\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2 / ||\boldsymbol{\epsilon}||_2^2$ as:

$$
\begin{aligned}
\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2 / ||\boldsymbol{\epsilon}||_2^2 &= \frac{\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2}{||\sum_{k=1}^d v_k v_k^T\boldsymbol{\epsilon}||_2^2}, &(17)\\
&= \frac{\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2}{\sum_{k=1}^d \left(v_k^T\boldsymbol{\epsilon}\right)^2}, &(18)\\
&= \frac{X_i^2}{\sum_{k=1}^d X_k^2}. &(19)
\end{aligned}
$$

Let $Y_i \coloneqq \frac{X_i^2}{\sum_{j=1}^d X_j^2}$. Because $\forall j \in [d], X_j \coloneqq v_j^T \boldsymbol{\epsilon} \sim \mathcal{N}(0,1), \forall j \in [d], Y_j$ has the same distribution. Additionally, $\sum_{j=1}^d Y_j = 1$. So:

$$\mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)}[\frac{\left(\boldsymbol{v}_i^T \boldsymbol{\epsilon}\right)^2}{||\boldsymbol{\epsilon}||_2^2}] = \mathbb{E}[Y_i] = \frac{1}{d}\mathbb{E}[\sum_{j=1}^d Y_j] = \frac{1}{d}.$$

□

Lemma 2. *Given a matrix $\boldsymbol{J} \in \mathbb{R}^{d \times d}$ with $\texttt{rank}\,(\boldsymbol{J}) = r$. Given $\boldsymbol{x}$ which is uniformly sampled on the unit hypersphere $\mathbb{S}^{d-1}$, we have:*

$$\mathbb{E}_{\boldsymbol{x}}\left[||\boldsymbol{J}\boldsymbol{x}||_2^2\right] = \frac{||\boldsymbol{J}||_F^2}{d}.$$

Proof of Lemma 2. Let's define the singular value decomposition of $\boldsymbol{J} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ with $\Sigma = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_r, 0 \ldots, 0)$. Therefore, $\mathbb{E}_{\boldsymbol{x}}\left[||\boldsymbol{J}\boldsymbol{x}||_2^2\right] = \mathbb{E}_{\boldsymbol{x}}\left[||\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x}||_2^2\right] = \mathbb{E}_{\boldsymbol{z}}\left[||\boldsymbol{\Sigma}\boldsymbol{z}||_2^2\right]$ where $\boldsymbol{z} \coloneqq \boldsymbol{V}^T\boldsymbol{x}$ is is uniformly sampled on the unit hypersphere $\mathbb{S}^{d-1}$. Thus, we have:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{z}}\left[||\boldsymbol{\Sigma}\boldsymbol{z}||_2^2\right] &= \mathbb{E}_{\boldsymbol{z}}\left[||\sum_{i=1}^r \sigma_i \boldsymbol{e}_i^T\boldsymbol{z}||_2^2\right], \\
&= \mathbb{E}_{\boldsymbol{z}}\left[\sum_{i=1}^r \sigma_i^2 ||\boldsymbol{e}_i^T\boldsymbol{z}||_2^2\right], \\
&= \sum_{i=1}^r \sigma_i^2 \mathbb{E}_{\boldsymbol{z}}\left[||\boldsymbol{e}_i^T\boldsymbol{z}||_2^2\right] = \frac{||\boldsymbol{J}||_F^2}{d},
\end{aligned}$$

where $\boldsymbol{e}_i$ is the standard basis with i-th element equals to 0. The second equality is because of independence between $\boldsymbol{e}_i^T\boldsymbol{z}$ and $\boldsymbol{e}_j^T\boldsymbol{z}$. The fourth equality is from Lemma 1. □

Lemma 3. *Given function $f\,(\boldsymbol{x}) = ||\boldsymbol{J}\boldsymbol{x}||_2^2$, the lipschitz constant L_f of function $f\,(\boldsymbol{x})$ is:*

$$L_f = 2||\boldsymbol{J}||_2^2.$$

Proof of Lemma 3. The jacobian of $f(\boldsymbol{x})$ is:

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = 2\boldsymbol{J}^T\boldsymbol{J}\boldsymbol{x},$$

Therefore, the lipschitz constant L follows:

$$L_f = \sup_{\boldsymbol{x} \in \mathbb{S}^{d-1}} ||\nabla_{\boldsymbol{x}} f(\boldsymbol{x})||_2 = 2 \sup_{\boldsymbol{x} \in \mathbb{S}^{d-1}} ||\boldsymbol{J}^T\boldsymbol{J}\boldsymbol{x}||_2 = ||\boldsymbol{J}^T\boldsymbol{J}||_2 = ||\boldsymbol{J}||_2^2$$

□

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 paper contributes a watermark for diffusion models, with the method introduced in Section 3, theoretical justification in Section 4 and experimental verification in Section 5.

   Guidelines:

   - The answer NA 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 NA 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: We carefully discuss the assumption in Section 4.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: We carefully discuss the assumption in Section 4 and attach proof in Appendix G.

Guidelines:

- The answer NA 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: We discuss detailed experiment settings in Section 5.

   Guidelines:

   - The answer NA 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: [No]

Justification: We will make it open source in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We discuss the detailed experiment settings in Section 5.

   Guidelines:

   - The answer NA 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: Due to computation limits, we are unable to run experiments multiple times.

   Guidelines:

   - The answer NA 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: All experiments are run over one A40 GPU.

   Guidelines:

   - The answer NA 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: We conduct the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: We have discussed in Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: We have cited the assets we used in the paper.

    Guidelines:

    - The answer NA 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: [No]

    Justification: We will provide the documentation when releasing the code.

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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: [NA]

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

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.