SHALLOW DIFFUSE: ROBUST AND INVISIBLE WATERMARKING THROUGH LOW-DIMENSIONAL SUBSPACES IN DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

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 our *Shallow Diffuse* outperforms existing watermarking methods in terms of robustness and consistency.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021b) have recently become a new dominant family of generative models, powering various commercial applications such as Stable Diffusion (Rombach et al., 2022; Esser et al., 2024), DALL-E (Ramesh et al., 2022; Betker et al., 2023), Imagen (Saharia et al., 2022) Stable Audio (Evans et al., 2024) and Sora (Brooks et al., 2024). 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 (Zellers et al., 2019; Goldstein et al., 2023; Brundage et al., 2018); (b) the memorization of training data by those models (Gu et al., 2023; Somepalli et al., 2023a;b; Wen et al., 2023b; Zhang et al., 2024a) challenges the originality of the generated content and raises potential copyright infringement issues; (c) Iterative training on AI-generated content, known as model collapse (Fu et al., 2024; Alemohammad et al., 2024; Dohmatob et al., 2024; Shumailov et al., 2024; Gibney, 2024) 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 to the image during the generation process; and (b) *the user scenario*: where given a generated image, the watermark is injected in a post-process manner; (as shown in the left two blocks in Figure 3). Traditional watermarking methods (Cox et al., 2007; Solachidis & Pitas, 2001; Chang et al., 2005; Liu et al., 2019) are mainly designed for the user scenario, embedding detectable watermarks directly into images with minimal modification. However, these methods are vulnerable 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 (Zhang et al., 2024c; Fernandez et al., 2023; Wen et al., 2023a; Yang et al., 2024; Ci et al., 2024), where they improve robustness by integrating watermarking into the sampling process of diffusion models. For example, the work (Ci et al., 2024;

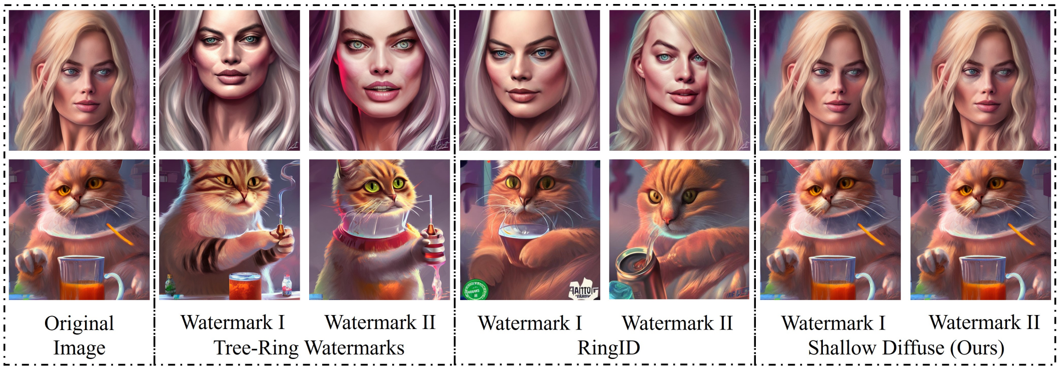


Figure 1: **Sampling variance of Tree-Ring Watermarks, RingID and Shallow Diffuse.** On the left are the original images, and on the right are the corresponding watermarked images generated using three different techniques: Tree-Ring (Wen et al., 2023a), RingID (Ci et al., 2024), and Shallow Diffuse. For each technique, we generated watermarks using two distinct random seeds, resulting in the respective watermarked images.

Wen et al., 2023a) embeds the watermark into the initial random seed in the Fourier domain and then samples an image from the watermarked seed. As illustrated in Figure 1, these approaches often lead to inconsistent watermarked images because they significantly alter the noise distribution away from Gaussian. Moreover, they require access to the initial random seed, limiting their use in the user scenario. To the best of our knowledge, there is currently no robust and consistent watermarking method suitable for both the server and user scenarios (more detailed discussion about related works could be found in Appendix A).

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. Unlike prior works (Ci et al., 2024; Wen et al., 2023a) that embed watermarks into the initial random seed and entangle the watermarking process with sampling, Shallow Diffuse decouples these two steps by leveraging the low-dimensional subspace in the generation process of diffusion models (Wang et al., 2024; Chen et al., 2024). The key insight is that, due to the low dimensionality of the subspace, a significant portion of the watermark will lie in the null space of this subspace, effectively separating the watermarking from the sampling process (see Figure 3 for an illustration). Our theoretical and empirical analyses demonstrate that this decoupling strategy significantly improves the consistency of the watermark. With better consistency as well as independence from the initial random seed, Shallow Diffuse is flexible for both server and user scenarios.

Our contributions. The proposed Shallow Diffuse offers several key advantages over existing watermarking techniques (Cox et al., 2007; Solachidis & Pitas, 2001; Chang et al., 2005; Liu et al., 2019; Zhang et al., 2024c; Fernandez et al., 2023; Wen et al., 2023a; Yang et al., 2024; Ci et al., 2024) 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 a straightforward 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 higher robustness and better consistency. Extensive experiments (Table 1 and Table 2) support our claims, with extra ablation studies in Figure 5a and Figure 5b .

- **Provable Guarantees.** Unlike previous methods, 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 (Ho et al., 2020; Song et al., 2021b; Karras et al., 2022), followed by several key empirical properties that will be used in our approach: the low-rankness and local linearity of the diffusion model (Wang et al., 2024; Chen et al., 2024).

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 Ho et al. (2020), 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 Ho et al. (2020); Song et al. (2021a); Lu et al. (2022a); Karras et al. (2022) 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) Song et al. (2021a) 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 Zhang et al. (2024b); Luo (2022), 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 $\boldsymbol{x}_{t_2} = \texttt{DDIM-Inv}(\boldsymbol{x}_{t_1}, t_2)$.

Text-to-image (T2I) diffusion models & classifier-free guidance (CFG). The diffusion model can be generalized from unconditional to T2I (Rombach et al., 2022; Esser et al., 2024), 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) (Ho & Salimans, 2022), 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, denotes as $\boldsymbol{x}_{t_1} = \texttt{DDIM}(\boldsymbol{x}_{t_2}, t_1, \boldsymbol{c})$ and $\boldsymbol{x}_{t_2} = \texttt{DDIM-Inv}(\boldsymbol{x}_{t_1}, t_2, \boldsymbol{c})$.

2.2 LOCAL LINEARITY AND INTRINSIC LOW-DIMENSIONALITY IN PMP

In this work, we will 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 (Wang et al., 2024; Manor & Michaeli, 2024b;a), and they have been extensively studied in (Chen et al., 2024). At one given timestep $t \in [0, T]$, let us 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 $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ is the Jacobian of $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$. As shown in (Chen et al., 2024), it has

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**
  $\boldsymbol{x}_t = \texttt{DDIM-Inv}\left(\boldsymbol{x}_0, t\right)$
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
$\boldsymbol{x}_0^{*} \leftarrow \texttt{DDIM}\left(\boldsymbol{x}_t, 0\right)$, $\boldsymbol{x}_0^{*\mathcal{W}} \leftarrow \texttt{ChannelAverage}\left(\boldsymbol{x}_0^{\mathcal{W}}, \boldsymbol{x}_0^{*}\right)$ ▷ Channel Average
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,
$\bar{\boldsymbol{x}}_t^{\mathcal{W}} \leftarrow \texttt{DDIM-Inv}\left(\bar{\boldsymbol{x}}_0^{\mathcal{W}}, t\right)$
$\eta = \texttt{Detector}\left(\bar{\boldsymbol{x}}_t^{\mathcal{W}}, \lambda\Delta\boldsymbol{x}\right)$
Return: η

been found that 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:

- **Low-rankness of the Jacobian $\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$.** As shown in Figure 2(a) of (Chen et al., 2024), 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 less than 10^{-2}) for all diffusion models within the range $t \in [0.2T, 0.7T]$, (*iii*) it achieves the lowest value around mid-to-late timestep, slightly differs on different architectures and datasets.
- **Local linearity of the PMP $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$.** As shown in Figure 2(b) of (Chen et al., 2024), the mapping $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ exhibits strong linearity across a large portion of the timesteps, which is consistently true among different architectures trained on different datasets. In particular, the work (Chen et al., 2024) evaluated the linearity of $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ at $t = 0.7T$ where the rank ratio is close to the lowest value, showing that $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda\Delta\boldsymbol{x}) \approx \boldsymbol{l}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \lambda\Delta\boldsymbol{x})$ even when $\lambda = 40$,

3 Watermarking by Shallow-Diffuse

In this section, we introduce Shallow Diffuse for watermarking diffusion models. Building on the benign properties of PMP discussed in Section 2.2, we explain 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 extend this approach to *text-to-image* diffusion models, illustrated in Figure 3.

3.1 Injecting invisible watermarks

Consider an unconditional diffusion model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ as we 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, $\boldsymbol{x}_t = \texttt{DDIM-Inv}\left(\boldsymbol{x}_0, t\right)$ 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

$$\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t^{\mathcal{W}}) \;=\; \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \lambda\underbrace{\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\cdot\Delta\boldsymbol{x}}_{\approx\boldsymbol{0}} \;\approx\; \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \;=\; \hat{\boldsymbol{x}}_{0,t}, \tag{4}$$

where we select the watermark $\Delta \boldsymbol{x}$ 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 $r_t = \operatorname{rank}(\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)) \ll d$ 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 ensuring that $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ is locally linear and that the dimensionality of its Jacobian $r_t \ll d$. In practice, we choose $t = 0.3T$ based on results from the ablation study in Section 5.4. As a results, the injection in Equation (4) maintains better consistency without changing the predicted $\boldsymbol{x}_0$. In the meanwhile, it is very robust because any attack on $\boldsymbol{x}_0$ would remain disentangled from the watermark, so that $\lambda \Delta \boldsymbol{x}$ remains detectable.

Although in practice we employ the DDIM method instead of PMP for sampling high-quality images, the above intuition still carries over to DDIM. From Equation (1), one step sampling of DDIM in terms of $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$ becomes:

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)}_{\text{"predicted } \boldsymbol{x}_0\text{"}} + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_t}} \underbrace{\left(\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\right)}_{\text{"the direction pointing to } \boldsymbol{x}_t\text{"}} . \tag{5}$$

As explained in Song et al. (2021a), the first term predicts $\boldsymbol{x}_0$ while the second term points towards $\boldsymbol{x}_t$. When we inject the watermark $\Delta \boldsymbol{x}$ into $\boldsymbol{x}_t$ as given in Equation (3), we know that

$$\begin{aligned}\boldsymbol{x}_{t-1}^{\mathcal{W}} &= \sqrt{\alpha_{t-1}}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t^{\mathcal{W}}) + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_t}}\left(\boldsymbol{x}_t^{\mathcal{W}} - \sqrt{\alpha_t}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t^{\mathcal{W}})\right) \\ &\approx \sqrt{\alpha_{t-1}}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \frac{\sqrt{1-\alpha_{t-1}}}{\sqrt{1-\alpha_t}}\left(\boldsymbol{x}_t + \lambda\Delta\boldsymbol{x} - \sqrt{\alpha_t}\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)\right),\end{aligned} \tag{6}$$

where the second 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 (6) and it decouples from the first which predicts $\boldsymbol{x}_0$. Therefore, similar to our analysis for PMP, the first term in equation 6 maintains the consistency of data generation, while the difference in second term highlighted by blue would be useful for detecting the watermark which we will discuss next. In Section 4, we provide more 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 Wen et al. (2023a), 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:

$$\lambda\Delta\boldsymbol{x} \;:=\; \texttt{DFT-Inv}\left(\texttt{DFT}\left(\boldsymbol{x}_t\right) \odot (1-\boldsymbol{M}) + \boldsymbol{W} \odot \boldsymbol{M}\right) - \boldsymbol{x}_t, \tag{7}$$

where the Hadamard product $\odot$ denotes the element-wise multiplication. Additionally, we have the following for Equation (7):

- **Transformation into the frequency domain.** Let $\texttt{DFT}(\cdot)$ and $\texttt{DFT-Inv}(\cdot)$ represent the forward and inverse Discrete Fourier Transform (DFT) operators, respectively. As shown in Equation (7), we first apply $\texttt{DFT}(\cdot)$ to transform $\boldsymbol{x}_t$ into the frequency domain, where we then introduce the watermark via a mask. Finally, the modified input is transformed back into the pixel domain using $\texttt{DFT-Inv}(\cdot)$.
- **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 2, 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 2, where we use it to modify specific frequency bands of the image. Specifically, the radius of the circle in mask M is 8, In the following, we discuss the design of $\boldsymbol{M}$ and $\boldsymbol{W}$ in detail.

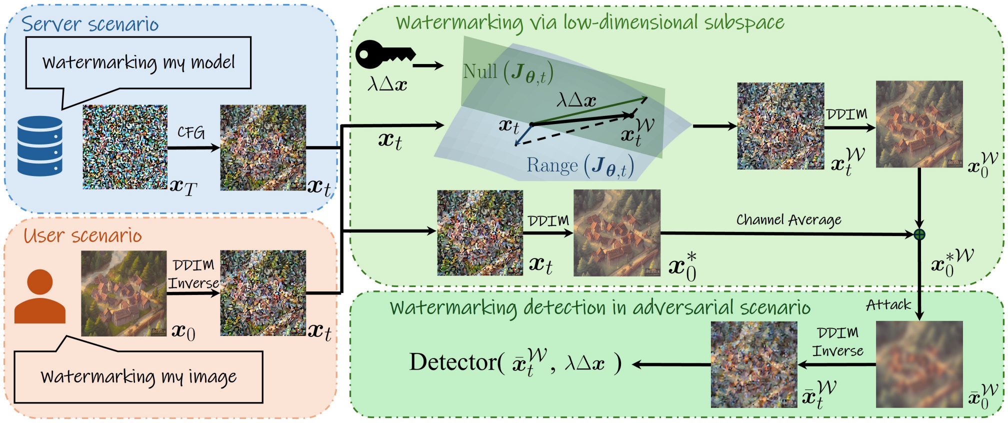


Figure 3: **Overview of Shallow Diffuse for T2I diffusion models.**

Previous methods (Wen et al., 2023a; Ci et al., 2024) design the mask $\boldsymbol{M}$ to modify the low-frequency components of the initial noise input. While this approach works, as most of the energy in natural images is concentrated in the low-frequency range, it tends to distort the image when such watermarks are injected (see Figure 1 for an illustration). In contrast, as shown in Figure 2, we design the mask $\boldsymbol{M}$ to target the high-frequency components of the image. Since high-frequency components capture fine details where the energy is less concentrated on these bands, modifying them results in less distortion of the original image. This is especially true in our case because we are modifying $\boldsymbol{x}_t$, which is closer to $\boldsymbol{x}_0$, compared to the initial noise used in (Wen et al., 2023a; Ci et al., 2024).To modify the high-frequency components, we apply the `DFT` without shifting and centering the zero frequency, as illustrated in the bottom-left of Figure 2.

In terms of designing the key $\boldsymbol{W}$, we follow Wen et al. (2023a). The key $\boldsymbol{W}$ is composed of multi-rings and each ring has the same value that is drawn from Gaussian distribution; see the top-right of Figure 2 for an illustration. Further ablation studies on the choice of $\boldsymbol{M}$, $\boldsymbol{W}$, and the effects of selecting low-frequency or high-frequency regions for watermarking can be found in Table 7.

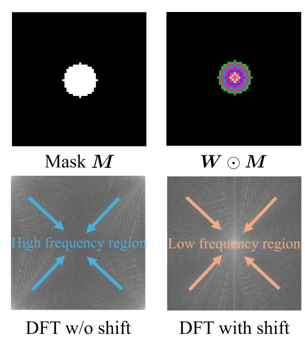


Figure 2: **Illustration of watermark patterns**.

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 $\bar{\boldsymbol{x}}_t^{\mathcal{W}} = \texttt{DDIM-Inv}\left(\bar{\boldsymbol{x}}_0^{\mathcal{W}}, t\right)$. To detect the watermark, following Wen et al. (2023a); Zhang et al. (2024c), the `Detector`$(\cdot)$ in Algorithm 1 calculates the following p-value:

$$\eta = \frac{\texttt{sum}(\boldsymbol{M}) \cdot ||\boldsymbol{M} \odot \boldsymbol{W} - \boldsymbol{M} \odot \texttt{DFT}\left(\bar{\boldsymbol{x}}_t^{\mathcal{W}}\right)||_F^2}{||\boldsymbol{M} \odot \texttt{DFT}\left(\bar{\boldsymbol{x}}_t^{\mathcal{W}}\right)||_F^2}, \tag{8}$$

where `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 choosing a threshold η_0, non-watermarked images will have $\eta > \eta_0$ and watermarked images will have $\eta < \eta_0$. Theoretically, the derivation of the p-value η could be found in Zhang et al. (2024c).

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

Up to this point, our discussion has focused exclusively on unconditional diffusion models. Next, we demonstrate how our approach can be readily extended to text-to-image (T2I) diffusion models, which are predominantly used in practice.

Figure 3 provides an overview of our method for T2I diffusion models, which can be flexibly applied to both server and user scenarios. Specifically,

- **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 $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ 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 $\boldsymbol{x}_t = \texttt{DDIM-Inv}\left(\boldsymbol{x}_0, t, \varnothing\right)$, using an empty prompt $\varnothing$. Next, similar to Section 3.1, we apply DDIM to obtain the watermarked image $\boldsymbol{x}_0^{\mathcal{W}} = \texttt{DDIM}\left(\boldsymbol{x}_t^{\mathcal{W}}, 0, \varnothing\right)$ and channel averaging $\boldsymbol{x}_0^{*\mathcal{W}} \leftarrow \texttt{ChannelAverage}\left(\boldsymbol{x}_0^{\mathcal{W}}, \texttt{DDIM}(\boldsymbol{x}_t, 0)\right)$. The detailed discussion about channel averaging is in Appendix B.
- **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 $\bar{\boldsymbol{x}}_t^{\mathcal{W}} = \texttt{DDIM-Inv}\left(\bar{\boldsymbol{x}}_0^{\mathcal{W}}, t, \varnothing\right)$. We detect the watermark $\Delta\boldsymbol{x}$ in $\bar{\boldsymbol{x}}_t^{\mathcal{W}}$ by calculating η in Equation (8), 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 introduced in Section 3 for unconditional diffusion models. First, we make the following assumptions on the watermark and the diffusion model process.

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

- ***Linearity:*** *For any small 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 a L-Lipschiz continuous at every t:*
$$||\boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x})||_2 \leq L, \quad \forall\boldsymbol{x} \in \mathbb{R}^d.$$

It should be noted that our assumptions are mild. The L-Lipschitz continuity is a common assumption for analysis. The approximated linearity have been shown in (Chen et al., 2024) with the assumption of data distribution to be a mixture of low-rank Gaussians. Here, we assume the linearity to be exact for the ease of analysis, and it can be generalized to approximate linear case.

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 the PMP $\boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$.

Theorem 1 (Consistency of the watermarks)**.** *Suppose Assumption 1 holds and $\Delta\boldsymbol{x} \sim \mathrm{U}(\mathbb{S}^{d-1})$. Let us define $\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} := \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t + \lambda\Delta\boldsymbol{x})$, $\hat{\boldsymbol{x}}_{0,t} := \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)$. The ℓ_2-norm distance between $\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}}$ and $\hat{\boldsymbol{x}}_{0,t}$ can be bounded by:*
$$||\hat{\boldsymbol{x}}_{0,t}^{\mathcal{W}} - \hat{\boldsymbol{x}}_{0,t}||_2 \leq \lambda L h(r_t), \tag{9}$$
with probability at least $1 - r_t^{-1}$. Here, $h(r_t) = \sqrt{\frac{r_t}{d} + \sqrt{\frac{18\pi^3}{d-2}\log\left(2r_t\right)}}$.

Our Theorem 1 guarantees that adding the watermark $\lambda\Delta\boldsymbol{x}$ would only change the estimation by an amount of $\lambda L h(r_t)$ with a constant probability. In particular, when r_t is small, it 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 the practical use. On the other hand, in the following we show that the injected watermark can be detected based upon the second term in Equation (6).

Theorem 2 (Detectability of the watermarks)**.** *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 $\bar{\boldsymbol{x}}_t^{\mathcal{W}} = \texttt{DDIM-Inv}\left(\boldsymbol{x}_{t-1}^{\mathcal{W}}, t\right)$. The ℓ_2-norm distance between $\tilde{\boldsymbol{x}}_t^{\mathcal{W}}$ and $\boldsymbol{x}_t^{\mathcal{W}}$ can be bounded by:*
$$||\bar{\boldsymbol{x}}_t^{\mathcal{W}} - \boldsymbol{x}_t^{\mathcal{W}}||_2 \leq \lambda L\left(-g\left(\alpha_t, \alpha_{t-1}\right) + g\left(\alpha_{t-1}, \alpha_t\right)\left(1 - Lg\left(\alpha_t, \alpha_{t-1}\right)\right)\right) h(\max\{r_{t-1}, r_t\}) \tag{10}$$
with probability at least $1 - r_t^{-1} - r_{t-1}^{-1}$. Here, $g(x, y) := \frac{\sqrt{1-y}\sqrt{x} - \sqrt{1-x}\sqrt{y}}{\sqrt{1-x}}$, $\forall x, y \in (0, 1)$.

Here $-g(\alpha_t, \alpha_{t-1}) + g(\alpha_{t-1}, \alpha_t)(1 - Lg(\alpha_t, \alpha_{t-1}))$ is a small number under the α_t designed for variance preserving (VP) noise scheduler Ho et al. (2020) and $h(\max\{r_{t-1}, r_t\})$ is small when r_t is small. This indicates that the difference between $\bar{\boldsymbol{x}}_t^{\mathcal{W}}$ and $\boldsymbol{x}_t^{\mathcal{W}}$ is small when r_t 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 constant 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). Additionally, we compare Shallow Diffuse with other related works in the trade-off between robustness and consistency, as detailed in Section 5.3. Moreover, we investigate the effect of timestep t on both robustness and consistency, with results presented in Section 5.4. We further explore the multi-key identification experiments in Appendix C.2. Lastly, we provide an ablation study on watermark pattern design (Appendix C.3), channel averaging (Appendix C.4), watermarking embedded channel (Appendix C.5), and sampling method (Appendix C.6).

Baseline For the server scenario, we select the following non-diffusion-based method: DWtDct Cox et al. (2007), DwtDctSvd Cox et al. (2007), RivaGAN Zhang et al. (2019), StegaStamp Tancik et al. (2020); and diffusion-based method: Stable Signature Fernandez et al. (2023), Tree-Ring Watermarks Wen et al. (2023a), RingId Ci et al. (2024), and Gaussian Shading Yang et al. (2024). In the user scenario, we adopt the same baseline methods, except for Stable Signature and Gaussian Shading, as these methods are not suitable for this setting.

Datasets We use Stable Diffusion 2-1-base (Rombach et al., 2022) 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 Lin et al. (2014), WikiArt Tan et al. (2019), and DiffusionDB datasets Wang et al. (2022). The first two are real-world datasets, 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, Section 5.4, Appendix C, we use the server scenario and sample 100 images for evaluation.

Metric To evaluate image consistency under the user scenario, we use peak signal-to-noise ratio (PSNR) Jähne (2005), structural similarity index measure (SSIM) Wang et al. (2004), and Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. (2018), comparing watermarked images to their original counterparts. In the server scenario, we assess the generation quality of the watermarked images using Contrastive Language-Image Pretraining Score (CLIP-Score) Radford et al. (2021) and Fréchet Inception Distance (FID) Heusel et al. (2017). To evaluate robustness, we vary the threshold η_0 and 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. Robustness is evaluated both under clean conditions (no attacks) and with various attacks, including JPEG compression, Gaussian blurring, Gaussian noise, and color jitter, Resize and restore, Random drop, median blurring, diffusion purification Nie et al. (2022), VAE-based image compression models Cheng et al. (2020); Ballé et al. (2018) and stable diffusion-based image regeneration Zhao et al. (2023b). We report the average robustness of these attacks in the main paper. Detailed settings and experiment results of these attacks are provided in Appendix C.1.

5.1 Consistency and robustness under the server scenario

Table 1 compares the performance of Shallow Diffuse with other methods in the user scenario. For reference, we also apply stable diffusion to generate images from the same random seeds, without adding watermarks (referred to as "Stable Diffusion w/o WM" in Table 1). In terms of generation quality, Shallow Diffuse achieves the best FID score among the diffusion-based methods. Additionally, the FID and CLIP scores of Shallow Diffuse are very close to those of Stable Diffusion

Table 1: **Generation quality and watermark robustness under the server scenario.**

Method	Generation Quality		Watermark Robustness (AUC ↑/TPR@1%FPR↑)	
	CLIP-Score ↑	FID ↓	Clean	Adversarial Average
Stable Diffusion w/o WM	0.3286	25.56	-	-
DwtDct	0.3298	25.73	0.97/0.85	0.61/0.18
DwtDctSvd	0.3291	26.00	**1.00/1.00**	0.79/0.46
RivaGAN	0.3252	24.60	1.00/0.99	0.85/0.57
Stegastamp	0.3552	**24.59**	**1.00/1.00**	0.97/0.87
Stable Signature	0.3622	30.86	**1.00/1.00**	0.83/0.44
Tree-Ring Watermarks	0.3310	25.82	**1.00/1.00**	0.98/0.87
RingID	0.3285	27.13	**1.00/1.00**	**1.00/1.00**
Gaussian Shading	**0.3631**	26.17	**1.00/1.00**	**1.00/1.00**
Shallow Diffuse (ours)	0.3285	25.58	**1.00/1.00**	**1.00/1.00**

Table 2: **Generation consistency and watermark robustness under the user scenario.**

Dataset	Method	Generation Consistency			Watermark Robustness (AUC ↑/TPR@1%FPR↑)	
		PSNR ↑	SSIM ↑	LPIPS ↓	Clean	Adversarial Avg.
COCO	Stable Diffusion w/o WM	32.28	0.78	0.06	-	-
	DwtDct	37.88	0.97	**0.02**	0.98/0.83	0.61/0.19
	DwtDctSvd	38.06	**0.98**	**0.02**	**1.00/1.00**	0.79/0.48
	RivaGAN	**40.57**	**0.98**	0.04	**1.00/1.00**	0.87/0.61
	Stegastamp	31.88	0.86	0.08	**1.00/1.00**	0.96/0.83
	Tree-Ring Watermarks	28.22	0.51	0.41	**1.00/1.00**	0.99/0.93
	RingID	28.22	0.38	0.61	**1.00/1.00**	**1.00/0.99**
	Shallow Diffuse (ours)	32.11	0.77	0.06	**1.00/1.00**	1.00/0.98
DiffusionDB	Stable Diffusion w/o WM	33.42	0.85	0.03	-	-
	DwtDct	37.77	0.96	**0.02**	0.96/0.76	0.61/0.18
	DwtDctSvd	37.84	0.97	**0.02**	**1.00/1.00**	0.79/0.46
	RivaGAN	**40.6**	**0.98**	0.04	1.00/0.98	0.85/0.57
	Stegastamp	32.03	0.85	0.08	**1.00/1.00**	0.96/0.84
	Tree-Ring Watermarks	28.3	0.62	0.29	**1.00/1.00**	0.97/0.85
	RingID	27.9	0.21	0.77	**1.00/1.00**	**1.00/0.99**
	Shallow Diffuse (ours)	33.07	0.84	0.04	**1.00/1.00**	0.99/0.97
WikiArt	Stable Diffusion w/o WM	31.6	0.7	0.09	-	-
	DwtDct	38.84	0.97	0.02	0.96/0.75	0.60/0.18
	DwtDctSvd	39.14	0.98	0.02	**1.00/1.00**	0.78/0.48
	RivaGAN	**40.44**	**0.98**	**0.05**	**1.00/1.00**	0.87/0.60
	Stegastamp	31.62	0.85	0.09	**1.00/1.00**	0.95/0.75
	Tree-Ring Watermarks	28.24	0.53	0.34	**1.00/1.00**	0.97/0.92
	RingID	27.90	0.19	0.78	**1.00/1.00**	0.99/0.98
	Shallow Diffuse (ours)	31.4	0.68	0.10	**1.00/1.00**	**1.00/0.99**

w/o WM. This similarity arises because the watermarked distribution produced by Shallow Diffuse remains highly consistent with the original generation distribution. Regarding robustness, Shallow Diffuse outperforms all other methods. Although both Gaussian Shading and RingID exhibit comparable generation quality and robustness in the server scenario, they are less suitable for the user scenario. Specifically, Gaussian Shading embeds the watermark into $\boldsymbol{x}_T$, which is not accessible to the user, while RingID suffers from poor consistency, as demonstrated in Figure 1 and Table 2.

5.2 Consistency and Robustness under the User Scenario

Table 2 presents a comparison of Shallow Diffuse's performance against other methods in the user scenario. 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 $\hat{x}_0 = \texttt{DDIM}(\texttt{DDIM-Inv}(\boldsymbol{x}_0, t, \varnothing), 0, \varnothing)$, and measure the data consistency between $\hat{x}_0$ and $\boldsymbol{x}_0$ (denotes in Stable Diffusion 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 8, Shallow Diffuse also demonstrates strong generation consistency. As for the robustness, Shallow Diffuse is comparable to RingID and outperforms all other methods in all three datasets. While RivaGAN achieves the best image consistency and comparable watermark robustness to Shallow Diffuse in the user scenario, Shallow Diffuse is much more efficient. Unlike RivaGAN, which requires training for each individual image, Shallow Diffuse only involves the computational overhead of DDIM and DDIM inversion.

5.3 Trade-off between Consistency and Robustness

Figure 4 illustrates the trade-off between consistency and robustness for Shallow Diffuse and other baselines. As the radius of $\boldsymbol{M}$ increases, the watermark intensity λ also increases, reducing image

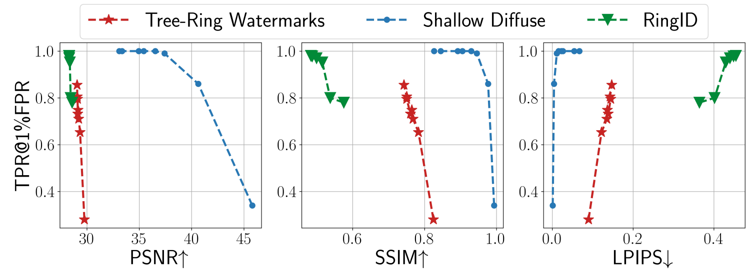
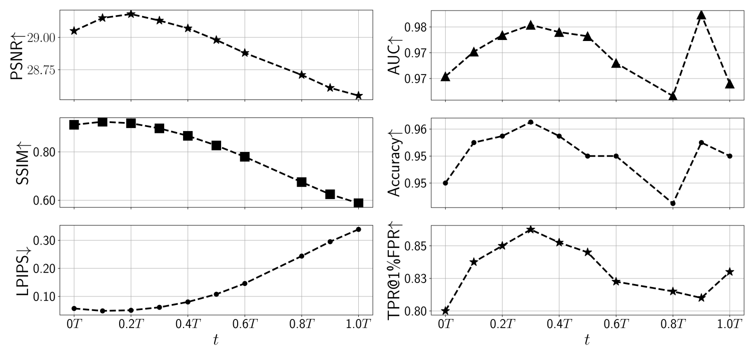


Figure 4: **Trade-off between consistency and robustness for Tree-Ring Watermarks, RingID, and Shallow Diffuse.**



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

Figure 5: **Ablation study of the watermark at different timestep** t**.**

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 4, 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.

5.4 Relation between injecting timestep, consistency and robustness

Figure 5 shows the relationship between the watermark injection timestep t and both consistency and robustness [1]. 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 Chen et al. (2024), 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 `DDIM-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 rank curve in Chen et al. (2024) is evaluated for an image-space diffusion model, whereas Shallow Diffuse operates in the latent-space diffusion model (e.g., Stable Diffusion).

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.

[1] 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 Watermarks.

REFERENCES

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.

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

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. URL `https://openreview.net/forum?id=ShjMHfmPs0`.

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. URL `https://openreview.net/forum?id=rkcQFMZRb`.

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.

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. URL `https://openai.com/research/video-generation-models-as-world-simulators`.

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.

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

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.

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*, pp. 7939–7948, 2020.

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.

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

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. URL `https://openreview.net/forum?id=KVvku47shW`.

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.

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.

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*, pp. 22466–22477, 2023.

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. URL https://openreview.net/forum?id=aw6L8sB2Ts.

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

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.

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

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.

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

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

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

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.

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.

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.

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*, pp. 740–755. Springer, 2014.

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.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PlKWVd2yBkY.

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. ISSN 2835-8856. URL https://openreview.net/forum?id=a90WpmSi0I. Survey Certification, Expert Certification.

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, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=2uAaGwlP_V.

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

Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=adSGeugiuj.

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 (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34603–34629. PMLR, 21–27 Jul 2024b. URL https://proceedings.mlr.press/v235/manor24a.html.

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)*, pp. 271–274. IEEE, 2008.

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.

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

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*, pp. 8748–8763. PMLR, 2021.

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.

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*, pp. 10684–10695, 2022.

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.

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.

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

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*, pp. 6048–6058, 2023a.

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*, 2023b. URL https://openreview.net/forum?id=HtMXRGbUMt.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

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*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

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. URL https://openreview.net/forum?id=a0XiA6v256.

Wei Ren Tan, Chee Seng Chan, Hernan Aguirre, and Kiyoshi Tanaka. Improved artgan for conditional synthesis of natural image and artwork. *IEEE Transactions on Image Processing*, 28(1): 394–409, 2019. doi: 10.1109/TIP.2018.2866698. URL https://doi.org/10.1109/TIP.2018.2866698.

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*, pp. 2117–2126, 2020.

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.

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.

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. URL https://arxiv.org/abs/2210.14896.

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*, 2023a. URL https://openreview.net/forum?id=Z57JrmubNl.

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*, 2023b.

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*, pp. 12162–12171, 2024.

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.

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*, 2024a.

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*, 2024b. URL https://openreview.net/forum?id=HsliOqZkc0.

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

Lijun Zhang, Xiao Liu, Antoni Viros Martin, Cindy Xiong Bearfield, Yuriy Brun, and Hui Guan. Robust image watermarking using stable diffusion, 2024c. URL https://arxiv.org/abs/2401.04247.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Loek7hfb46P.

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*, pp. 586–595, 2018.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *NeurIPS*, 2023a. URL http://papers.nips.cc/paper_files/paper/2023/hash/9c2aa1e456ea543997f6927295196381-Abstract-Conference.html.

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

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

A RELATED WORK

A.1 IMAGE WATERMARKING

Image watermarking has long been a crucial method for protecting intellectual property in computer vision (Cox et al., 2007; Solachidis & Pitas, 2001; Chang et al., 2005; Liu et al., 2019). Traditional techniques primarily focus on user-side watermarking, where watermarks are embedded into images post-generation. These methods (Al-Haj, 2007; Navas et al., 2008) 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 (Zhang et al., 2024c; Fernandez et al., 2023; Ahmadi et al., 2020; Lee et al., 2020; Zhu et al., 2018) 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 Wen et al. (2023a); Yang et al. (2024); Ci et al. (2024) 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 Wen et al. (2023a); Ci et al. (2024) significantly alter the data distribution, leading to variance towards watermarks in generated outputs (as shown in Figure 1).

In contrast to Wen et al. (2023a); Ci et al. (2024), our proposed shallow diffuse disentangles the watermark embedding from the generation process by leveraging the high-dimensional null space. This approach, both empirically and theoretically validated, significantly improves watermark consistency and robustness. To the best of our knowledge, this is the first method that supports watermark embedding for both server-side and user-side applications while maintaining high robustness and consistency.

A.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 (Loaiza-Ganem et al., 2024). This hypothesis suggests that high-dimensional real-world data actually lies in latent manifolds with a low intrinsic dimension. Focusing on diffusion models, Stanczuk et al. (2024) 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, Wang et al. (2024); Chen et al. (2024) find that the estimated posterior mean from diffusion models lies within this low-dimensional space. Additionally, Chen et al. (2024) 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, Stanczuk et al. (2024); Kamkari et al. (2024) introduce a local intrinsic dimension estimator, while Loaiza-Ganem et al. (2024) proposes a method for detecting out-of-domain data. Wang et al. (2024) offers theoretical insights into how diffusion model training transitions from memorization to generalization, and Chen et al. (2024); Manor & Michaeli (2024b) 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.

B Channel averaging

B.1 Technique details

Natural images have multiple channels denoted by C. Instead of applying watermark $\lambda\Delta$ to all channels of $\boldsymbol{x}_t$, we can apply the watermark to a specific channel c to make it even more invisible and robust. For this consideration, let us reshape the image $\boldsymbol{x}_t$ and the watermark $\Delta\boldsymbol{x}$ into the form $\boldsymbol{x}_t \in \mathbb{R}^{H\times W\times C}, \lambda\Delta\boldsymbol{x} \in \mathbb{R}^{H\times W\times C}$, where H, W, and C represent the height, width, and channel dimensions for the image, respectively. These dimensions satisfy $HWC = d$.

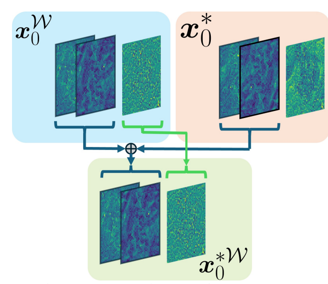


Figure 6: Illustration of channel average

Denote $[\boldsymbol{x}_t]_i \in \mathbb{R}^{H\times W}$ as the ith channel of $\boldsymbol{x}_t$, with $i \in [C]$. Thus $[\boldsymbol{x}_t^{\mathcal{W}}]_c = [\boldsymbol{x}_t]_c + [\lambda\Delta\boldsymbol{x}]_c$ and $[\boldsymbol{x}_t^{\mathcal{W}}]_i = [\boldsymbol{x}_t]_i$ for $i \neq c$. For the watermark in Equation (3), the channel averaging is defined as:

$$[\boldsymbol{x}_0^{*\mathcal{W}}]_i = \texttt{ChannelAverage}\left(\boldsymbol{x}_0^{\mathcal{W}}, \boldsymbol{x}_0^{*}\right), \tag{11}$$

$$= \begin{cases} [\boldsymbol{x}_0^{\mathcal{W}}]_i, i = c \\ (1-\gamma)[\boldsymbol{x}_0^{\mathcal{W}}]_i + \gamma[\boldsymbol{x}_0^{*}]_i, i \neq c \end{cases}, \tag{12}$$

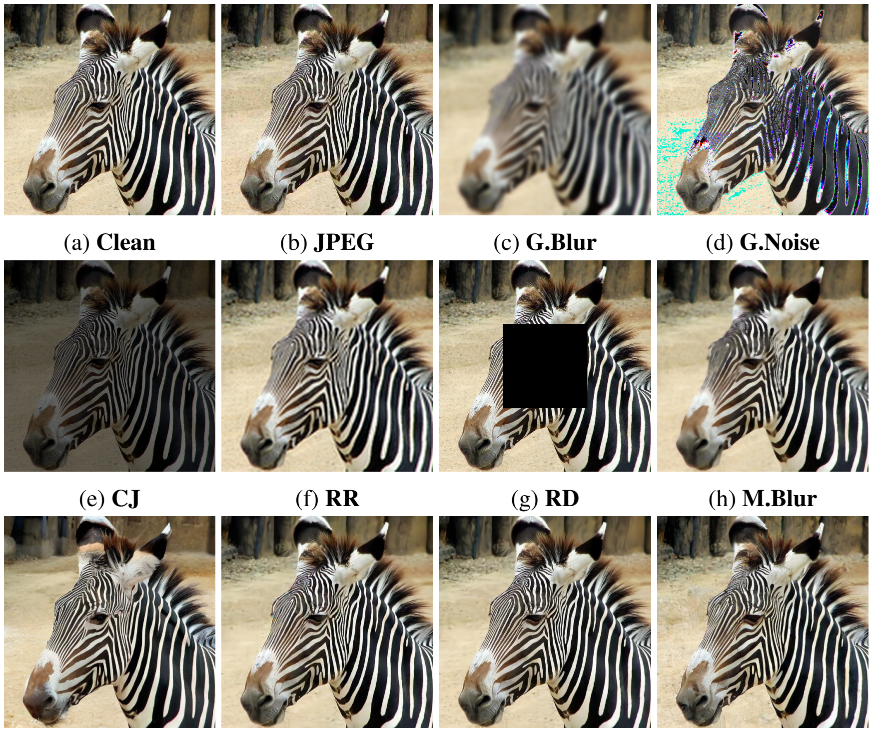
where we applied $\gamma = 1$. In our experiments, we found that we can increase both imperceptibility and robustness by further employing this simple approach. See our ablation study in Appendix C.4 for a more detailed analysis.

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 represent the most common image distortion methods in real life, including

- 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.



(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**

Figure 7: **Visualization of different attacks.**

- 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.
- Diffusion purification Nie et al. (2022) (DiffPure) with the purified step at 0.3T.
- VAE-based image compression Cheng et al. (2020) (IC1) and Ballé et al. (2018) (IC2), with a quality level of 3.
- Diffusion-based image regeneration Zhao et al. (2023b) with 60 denoising steps.

Visualizations of these attacks are in Figure 7. Detailed experiments for table 1 (Table 4) on above attacks are in Table 3 (Table 4).

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

Method	Watermarking Robustness (AUC ↑/TPR@1%FPR↑)												
	Clean	JPEG	G.Blur	G.Noise	CJ	RR	RD	M.Blur	DiffPure	IC1	IC2	IR	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	0.50/0.00	0.52/0.01	0.49/0.00	0.50/0.00	0.61/0.18
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**	0.51/0.02	0.73/0.03	0.68/0.04	0.70/0.07	0.79/0.46
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	0.73/0.16	0.65/0.03	0.63/0.04	0.56/0.00	0.85/0.57
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**	0.81/0.29	1.00/0.97	1.00/0.99	0.90/0.43	0.97/0.87
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	0.54/0.01	0.93/0.58	0.91/0.50	0.67/0.02	0.83/0.44
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	0.98/0.73	0.99/0.97	0.99/0.98	0.99/0.92	0.98/0.87
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**	**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**	**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**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**

C.2 MULTI-KEY WATERMARKING

In this section, we examine the capability of Shallow Diffuse to support multi-key watermarking. We evaluate two important tasks associated with multi-key watermarking: Multi-key identification and Multi-key re-watermarking.

Table 4: **Watermarking Robustness for different attacks under the user scenario.**

Method	Watermarking Robustness (AUC ↑/TPR@1%FPR↑)												
	Clean	JPEG	G.Blur	G.Noise	CJ	RR	RD	M.Blur	DiffPure	IC1	IC2	IR	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	0.46/0.00	0.49/0.00	0.49/0.01	0.46/0.00	0.61/0.19
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**	0.50/0.01	0.70/0.05	0.64/0.04	0.68/0.07	0.79/0.48
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**	0.63/0.02	0.68/0.05	0.66/0.04	0.75/0.15	0.87/0.61
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**	0.81/0.27	1.00/0.95	1.00/0.95	0.85/0.28	0.96/0.83
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**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.99/0.93
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**	**1.00/1.00**	**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**	0.99/0.86	1.00/0.99	0.99/0.97	**1.00/1.00**	1.00/0.98
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	0.50/0.004	0.52/0.008	0.49/0.004	0.50/0.002	0.61/0.18
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**	0.51/0.022	0.73/0.03	0.68/0.04	0.70/0.07	0.79/0.46
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**	0.56/0.004	0.65/0.03	0.63/0.04	0.73/0.16	0.85/0.57
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	0.83/0.28	1.00/0.91	1.00/0.93	0.85/0.40	0.96/0.84
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**	0.99/0.99	0.99/0.99	0.99/0.98	0.96/0.92	0.97/0.85
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**	**1.00/1.00**	**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**	0.96/0.90	0.96/0.92	0.97/0.93	0.98/0.96	0.99/0.97
WikiArt Dataset													
DwtDct	0.96/0.75	0.46/0.004	0.51/0.008	0.95/0.75	0.50/0.13	0.68/0.13	0.98/0.87	0.61/0.08	0.48/0.006	0.47/0.006	0.49/0.002	0.48/0.006	0.60/0.18
DwtDctSvd	**1.00/1.00**	0.65/0.22	0.97/0.76	0.99/0.99	0.50/0.10	1.00/1.00	1.00/1.00	**1.00/1.00**	0.47/0.03	0.72/0.04	0.66/0.07	0.67/0.08	0.78/0.48
RivaGAN	**1.00/1.00**	0.96/0.80	0.99/0.95	0.98/0.93	0.89/0.66	1.00/1.00	1.00/1.00	**1.00/1.00**	0.63/0.02	0.66/0.04	0.67/0.04	0.80/0.11	0.87/0.60
Stegastamp	**1.00/1.00**	1.00/0.96	0.97/0.77	0.92/0.88	0.98/0.84	0.99/0.89	**1.00/1.00**	0.99/0.91	0.77/0.20	0.99/0.95	0.99/0.90	0.80/0/30	0.95/0.75
Tree-Ring Watermarks	**1.00/1.00**	1.00/0.97	1.00/0.88	**1.00/1.00**	0.71/0.26	**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**	0.97/0.92
RingID	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.95/0.82	0.99/0.98	**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**	0.99/0.98
Shallow Diffuse (ours)	**1.00/1.00**	1.00/0.99	1.00/0.99	**1.00/1.00**	**1.00/0.99**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	0.97/0.94	0.98/0.95	**1.00/1.00**	**1.00/0.99**

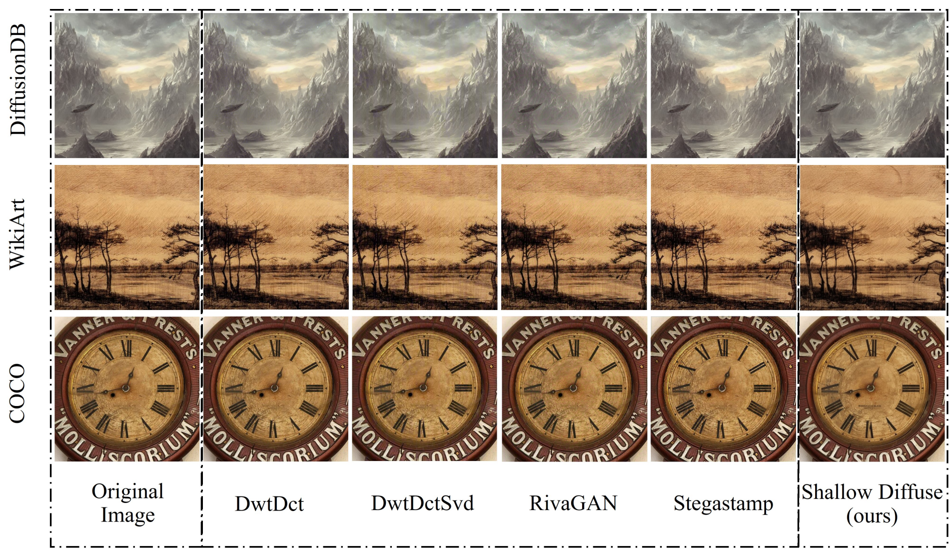


Figure 8: **Generation Consistency in User Scenarios.** We compare the visualization quality of our method against DwtDct, DwtdctSvd, RivaGAN, and Stegastamp across the DiffusionDB, WikiArt, and COCO datasets.

Multi-key identification This is a classification task designed to test the ability to accurately identify individual watermarks. We generate a set of $N = 2048$ watermarks, all using the same circular mask $\boldsymbol{M}$ but with distinct ring-shaped keys $\{\boldsymbol{W}_i\}_{i=1}^{N}$. During watermarking, a random key W_i is selected and injected into images. After an attack is applied, we attempt to detect the watermark key W_j and determine if $i = j$. The success rate of identification serves as the evaluation metric. This setup is inspired by the work in Ci et al. (2024). We compare Tree-Ring, RingID, and Shallow Diffuse in the server scenario. The results of this experiment are shown in Table 5. Despite lacking a dedicated design for multi-key scenarios, Shallow Diffuse outperforms Tree-Ring. RingID, specifically designed for multi-key identification, achieves the highest success rate. Exploring multi-key identification strategies could be an important direction for future research.

Multi-key re-watermarking : This task evaluates 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,

Original Image StageStamp Shallow Diffuse (ours)

Figure 9: **Generation Consistency in server scenarios.** We compare the visualization quality of our method against the original image and StageStamp.

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. 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 6. 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 5: **Multi-key identification for different attacks under the server scenario.**

Method	Successfull Rate ↑									
	Clean	JPEG	G.Blur	G.Noise	CJ	DiffPure	IC1	IC2	IR	Average
Tree-Ring	0.20	0.04	0.09	0.07	0.06	0.06	0.28	0.29	0.23	0.15
RingID	**1.00**	**0.97**	**0.97**	**0.95**	**0.87**	**0.88**	**0.98**	**0.98**	**0.99**	**0.95**
Shallow Diffuse	0.88	0.77	0.57	0.88	0.40	0.48	0.41	0.64	0.80	0.65

C.3 Ablation study of different watermark patterns

In Table 7, 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 2 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"

Table 6: **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

indicates a DFT without zero-frequency centering, as illustrated in Figure 2 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 7, 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 7: **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 Channel Average

We evaluate Shallow Diffuse with channel averaging enabled ($\gamma = 1.0$) and disabled ($\gamma = 0.0$), as shown in Table 8. Unlike the adaptive image enhancement techniques proposed in Zhang et al. (2024c), our approach embeds the watermark in a single channel while averaging the non-watermarked channels. This design takes advantage of the fact that many image processing operations, such as color jittering or Gaussian blurring, tend to affect all channels uniformly. By isolating the watermark in one channel, it will be less vulnerable to those attacks. Thus, applying channel averaging slightly enhances robustness against certain attacks while maintaining comparable consistency. Therefore, we set $\gamma = 1.0$ for Shallow Diffuse.

Table 8: **ablation study on channel average.**

Channel average intensity γ	PSNR ↑	SSIM ↑	LPIPS ↓	Watermarking Robustness (TPR@1%FPR↑)				
				Clean	JPEG	G.Blur	G.Noise	Color Jitter
0	**37.1103**	**0.941**	0.0154	**1.0000**	**1.0000**	0.9971	**1.0000**	0.9584
1.0	36.6352	0.931	**0.0151**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**

C.5 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 [2]. Since applying watermarking to any single channel yields similar results (Row

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

1-4), but applying it to all channels (Row 5) negatively impacts image consistency and robustness, we set $c = 3$ for Shallow Diffuse. This finding aligns with the observations in the channel average ablation study (appendix C.4). 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.6 Ablation study of different sampling methods

We conducted ablation studies on various diffusion model sampling methods, including DDIM, DEIS Zhang & Chen (2023), DPM-Solver Lu et al. (2022b), PNDM Liu et al. (2022), and UniPC Zhao et al. (2023a). All methods were evaluated using 50 sampling steps. The results, presented in Table 10, indicate that Shallow Diffuse is not highly sensitive to the choice of sampling method. Across all methods, the generation quality and watermark robustness remain consistent.

Table 10: **Ablation study on sampling methods.**

Sampling Method	Generation Quality	Watermark Robustness (AUC ↑/TPR@1%FPR↑)					
	CLIP-Score ↑	Clean	JPEG	G.Blur	G.Noise	CJ	Adversarial Average
DDIM	**0.3652**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**
DEIS	0.3651	**1.00/1.00**	0.99/0.99	**1.00/1.00**	**1.00/1.00**	0.99/0.95	1.00/0.99
DPM-Solver	0.3645	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	1.00/0.99	0.99/0.94	1.00/0.98
PNDM	0.3651	**1.00/1.00**	0.99/0.99	**1.00/1.00**	**1.00/1.00**	0.98/0.96	1.00/0.99
UniPC	0.3645	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**	**1.00/1.00**

D Proofs in Section 4

D.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 Popescu et al. (2006), 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 $1 - \frac{1}{r_t}$, 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:

$$
\begin{aligned}
\boldsymbol{x}_{t-1} &= \sqrt{\alpha_{t-1}} \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) + \sqrt{1-\alpha_{t-1}} \left(\frac{\boldsymbol{x}_t - \sqrt{\alpha_t} \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t)}{\sqrt{1-\alpha_t}} \right), && (13) \\
&= \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}} \boldsymbol{x}_t + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}} \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t), && (14)
\end{aligned}
$$

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 (14), we could obtain:

$$
\begin{aligned}
\boldsymbol{x}_{t-1}^{\mathcal{W}} &= \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}} \boldsymbol{x}_t^{\mathcal{W}} + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}} \boldsymbol{f}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t^{\mathcal{W}}), && (15) \\
&= \boldsymbol{x}_{t-1} + \sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}} \lambda \Delta \boldsymbol{x} + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}} \boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \Delta \boldsymbol{x} && (16) \\
&= \boldsymbol{x}_{t-1} + \lambda \underbrace{\left(\sqrt{\frac{1-\alpha_{t-1}}{1-\alpha_t}} \boldsymbol{I} + \frac{\sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}} - \sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t}}{\sqrt{1-\alpha_t}} \boldsymbol{J}_{\boldsymbol{\theta},t}(\boldsymbol{x}_t) \right)}_{:= \boldsymbol{W}_t} \Delta \boldsymbol{x}, && (17)
\end{aligned}
$$

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

$$
\boldsymbol{x}_t = \sqrt{\frac{1-\alpha_t}{1-\alpha_{t-1}}} \boldsymbol{x}_{t-1} + \frac{\sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t} - \sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}}}{\sqrt{1-\alpha_{t-1}}} \boldsymbol{f}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1}), \tag{18}
$$

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

$$
\begin{aligned}
\tilde{x}_t^{\mathcal{W}} &= \sqrt{\frac{1-\alpha_t}{1-\alpha_{t-1}}} \boldsymbol{x}_{t-1}^{\mathcal{W}} + \frac{\sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t} - \sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}}}{\sqrt{1-\alpha_{t-1}}} \boldsymbol{f}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1}^{\mathcal{W}}), \\
&= \boldsymbol{x}_t + \lambda \underbrace{\left(\sqrt{\frac{1-\alpha_t}{1-\alpha_{t-1}}} \boldsymbol{I} + \frac{\sqrt{1-\alpha_{t-1}}\sqrt{\alpha_t} - \sqrt{1-\alpha_t}\sqrt{\alpha_{t-1}}}{\sqrt{1-\alpha_{t-1}}} \boldsymbol{J}_{\boldsymbol{\theta},t-1}(\boldsymbol{x}_{t-1}) \right)}_{:= \boldsymbol{W}_{t-1}} \boldsymbol{W}_t \Delta \boldsymbol{x}, \\
&= \boldsymbol{x}_t + \lambda \boldsymbol{W}_{t-1} \boldsymbol{W}_t \Delta \boldsymbol{x} = \boldsymbol{x}_t^{\mathcal{W}} + \lambda \left(\boldsymbol{W}_{t-1} \boldsymbol{W}_t - \boldsymbol{I} \right) \Delta \boldsymbol{x}.
\end{aligned}
$$

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 $1 - \frac{1}{r_t} - \frac{1}{r_{t-1}}$,

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

□

E 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 $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$,

$$
\boldsymbol{v}_i^T\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{v}_i^T\boldsymbol{0}, \boldsymbol{v}_i^T\boldsymbol{I}_d\boldsymbol{v}_i) = \mathcal{N}(\boldsymbol{v}_i^T\boldsymbol{0}, \boldsymbol{v}_i^T\boldsymbol{I}_d\boldsymbol{v}_i) = \mathcal{N}(0, 1), \tag{19}
$$

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 $\forall j \in [d], X_j := v_j^T\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$. Therefore, we could rewrite $\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2 / ||\boldsymbol{\epsilon}||_2^2$ as:

$$
\begin{aligned}
\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2 / ||\boldsymbol{\epsilon}||_2^2 &= \frac{\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2}{|| \sum_{k=1}^d v_k v_k^T \boldsymbol{\epsilon}||_2^2}, &(20)\\
&= \frac{\left(\boldsymbol{v}_i^T\boldsymbol{\epsilon}\right)^2}{\sum_{k=1}^d \left(v_k^T\boldsymbol{\epsilon}\right)^2}, &(21)\\
&= \frac{X_i^2}{\sum_{k=1}^d X_k^2}. &(22)
\end{aligned}
$$

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 $\texttt{rank}\left(\boldsymbol{J}\right) = r$. 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 $\Sigma = \operatorname{diag}\left(\sigma_1, \dots, \sigma_r, 0 \dots, 0\right)$. 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 $f\left(\boldsymbol{x}\right) = ||\boldsymbol{J}\boldsymbol{x}||_2^2$, the lipschitz constant L_f of function $f\left(\boldsymbol{x}\right)$ 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:

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

□