
Adjusting Initial Noise to Mitigate Memorization in Text-to-Image Diffusion Models

Hyeonggeun Han^{1,2*} Sehwan Kim^{1*} Hyungjun Joo^{1,2}
Sangwoo Hong^{3†} Jungwoo Lee^{1,2,4†}

¹ECE & ²NextQuantum, Seoul National University

³CSE, Konkuk University

⁴Hodoo AI Labs

{hygnhan, sehwan kim, joo hj911, jungwoo lee}@snu.ac.kr
swhong06@konkuk.ac.kr

Abstract

Despite their impressive generative capabilities, text-to-image diffusion models often memorize and replicate training data, prompting serious concerns over privacy and copyright. Recent work has attributed this memorization to an attraction basin—a region where applying classifier-free guidance (CFG) steers the denoising trajectory toward memorized outputs—and has proposed deferring CFG application until the denoising trajectory escapes this basin. However, such delays often result in non-memorized images that are poorly aligned with the input prompts, highlighting the need to promote earlier escape so that CFG can be applied sooner in the denoising process. In this work, we show that the initial noise sample plays a crucial role in determining when this escape occurs. We empirically observe that different initial samples lead to varying escape times. Building on this insight, we propose two mitigation strategies that adjust the initial noise—either collectively or individually—to find and utilize initial samples that encourage earlier basin escape. These approaches significantly reduce memorization while preserving image-text alignment. Code is available at https://github.com/hygnhan/init_noise_diffusion_memorization.

1 Introduction

Text-to-image diffusion models have garnered significant attention for their remarkable ability to generate high-quality images that are semantically aligned with textual prompts. Despite their success, a growing body of work has revealed that these models are prone to memorizing and reproducing content from their training data, sometimes generating images that are nearly identical to those seen during training [39, 5, 40]. This memorization behavior poses serious concerns, including potential leakage of privacy-sensitive or copyrighted material [4, 23]. In addition to the legal and ethical risks, such behavior undermines the utility of diffusion models by limiting their generative diversity and reducing their capacity to synthesize truly novel content. These concerns highlight the pressing need to understand and mitigate memorization in diffusion models to ensure their safe and effective deployment.

A range of approaches have been proposed to mitigate memorization in diffusion models [49, 19, 34, 40]. Among them, inference-time mitigation strategies have gained particular attention due to their computational efficiency and practicality for real-world deployment. Several studies have linked

*These authors contributed equally to this work

†Corresponding authors

memorization to specific prompt-image associations, motivating mitigation methods that perturb prompts during inference [40, 49]. Other researches have focused on internal model behaviors, particularly within cross-attention layers, revealing that memorization can be attributed to sharply focused attention maps [34] or a small set of specific neurons [19]. Building on these insights, mitigation techniques such as attention logit rescaling [34] and neuron deactivation [19] have been introduced.

A recent line of work has identified classifier-free guidance (CFG) [20] in the early stages of denoising as a key factor contributing to memorization in diffusion models [22]. To explain this phenomenon, Jain et al. [22] introduced the concept of an *attraction basin*—a region in the sample-time space where applying CFG consistently drives the generation process toward memorized outputs, regardless of the random initialization. To mitigate this effect, the authors proposed deferring the use of CFG until the denoising trajectory exits the attraction basin. However, they also highlighted a trade-off: applying CFG too late can lead to non-memorized images that are poorly aligned with the conditioning prompt. To address this issue, they emphasized the need to escape the attraction basin earlier in the generation process. Toward that end, the authors introduced opposite guidance, a strategy that applies inverted conditional signals during early denoising steps to push the trajectory away from the attraction basin, thereby enabling earlier and safer application of CFG.

In this paper, we propose a novel approach to mitigating memorization by enabling an earlier escape from the attraction basin. In contrast to prior efforts that modify the denoising trajectory through adjustments to guidance signals, we instead focus on the starting point of the trajectory. Our empirical findings reveal that different initial Gaussian noise samples result in significantly different escape times from the attraction basin. This observation motivates the hypothesis that certain initializations naturally lie closer to the basin’s boundary, allowing the generation process to evade the memorization-inducing steering force more quickly. Building on this insight, we propose two inference-time mitigation strategies that find and utilize favorable initializations to promote early basin escape. Specifically, we propose a batch-wise method, which adjusts initial samples collectively, and a per-sample method, which adapts each initialization individually. Our approach offers a novel perspective that diverges from the conventional wisdom [49, 22] that the choice of initialization has minimal impact on memorization mitigation.

The key contributions of this paper are summarized as follows:

- We offer a novel perspective on the role of the initial noise sample in memorization mitigation, employing the concept of the attraction basin—a region in which classifier-free guidance (CFG) steers the denoising trajectory toward memorized outputs. We show that different initializations lead to varying escape times from this basin.
- We propose two inference-time mitigation strategies—Batch-wise and Per-sample approaches—that find and leverage initial samples likely to escape the attraction basin earlier. These methods adjust the initial samples to produce non-memorized images while promoting better alignment with the conditioning prompt.
- Our empirical evaluations demonstrate that the proposed initial sample adjustment strategies produce more text-aligned, non-memorized outputs than those without adjustment, and outperform existing inference-time mitigation baselines in reducing memorization.

2 Background

2.1 Diffusion models

Diffusion models [21, 42] aim to generate samples from a target data distribution $q(x)$. During training, a data sample $x_0 \sim q(x)$ is progressively corrupted by Gaussian noise over T timesteps such that $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. At each timestep t , the forward noising process is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where β_t denotes the variance schedule. The noise predictor ϵ_θ is trained to predict the noise added at each timestep. At inference time, the sampling process begins from $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and a sample is generated by iteratively denoising via the reverse process:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_\theta(x_t, t) \right), \tag{2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$.

Text-to-image diffusion models such as Stable Diffusion [35] usually utilize CFG to steer the generation process toward a desired conditioning input y . Given a null condition y_{null} representing the unconditional case, CFG modifies the noise prediction at each timestep as:

$$\epsilon^{\text{CFG}}(x_t, t, y) = \epsilon_\theta(x_t, t, y_{\text{null}}) + w_{\text{CFG}}(\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, y_{\text{null}})), \quad (3)$$

where w_{CFG} controls the strength of the guidance. For notational convenience, we denote the conditional noise prediction $\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, y_{\text{null}})$ as $\tilde{\epsilon}_\theta(x_t, t, y)$. It is worth noting that the trained noise predictor ϵ_θ approximates the score function at each timestep [20, 43]: $\epsilon_\theta(x_t, t, y) \approx -\sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log p_\theta(x_t | y)$ and $\epsilon_\theta(x_t, t, y_{\text{null}}) \approx -\sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log p_\theta(x_t)$.

2.2 Attraction basin

The attraction basin refers to a region formed in the denoising trajectory where applying CFG exerts a strong steering force toward memorized outputs. To mitigate memorization, Jain et al. [22] proposed deferring the application of CFG until the trajectory escapes this basin. A key aspect of this strategy is identifying when this escape occurs. They observed that when sampling proceeds without CFG, the magnitude of the conditional noise prediction $\|\tilde{\epsilon}_\theta(x_t, t, y)\|_2$ remains high within the basin. This is interpreted as empirical evidence of the memorization-inducing force present in this region. Just before the escape, the magnitude drops sharply—this timestep is referred to as the transition point. This behavior is illustrated in Figure 1. Applying CFG before the transition point (*i.e.*, within the attraction basin) typically results in memorized outputs, whereas applying it after the transition point leads to non-memorized images. However, excessively delaying CFG to avoid memorization may result in poor alignment with the conditioning prompt. This trade-off underscores the importance of encouraging earlier escape from the basin—*i.e.*, shifting the transition point to an earlier timestep—to generate non-memorized images that remain faithful to the prompt.

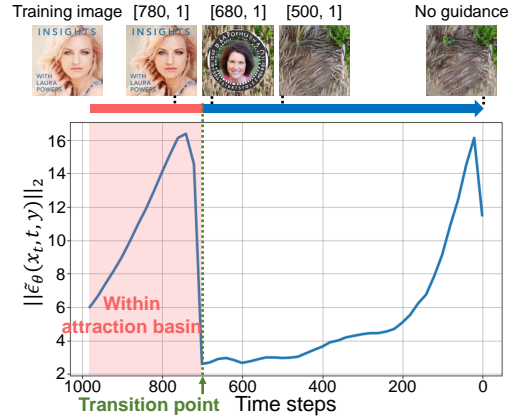


Figure 1: The magnitude of $\tilde{\epsilon}_\theta(x_t, t, y)$ at each timestep during sampling without CFG for a memorized prompt. Each image above corresponds to an output generated when CFG is applied over the timestep interval indicated by the associated square brackets.

Figure 1: The magnitude of $\tilde{\epsilon}_\theta(x_t, t, y)$ at each timestep during sampling without CFG for a memorized prompt. Each image above corresponds to an output generated when CFG is applied over the timestep interval indicated by the associated square brackets.

3 Initial sample adjustment for mitigating memorization

3.1 Motivation

To effectively mitigate memorization, CFG should be applied after the denoising trajectory escapes the attraction basin—*i.e.*, after the transition point. This motivates the need to shift the transition point to an earlier timestep. To this end, we investigate the influence of the initial Gaussian sample x_T , which defines the starting point of the trajectory. As shown in Figure 2, we consistently observe across different memorized prompts that varying x_T results in different transition points. This suggests that the location of x_T , relative to the attraction basin, influences how quickly the trajectory escapes. Based on this observation, we conjecture that initial samples closer to the basin’s boundary facilitate earlier escapes and thus earlier transition points. To validate this conjecture, we require a proxy for estimating a sample’s proximity to the basin’s boundary. Prior work [22] reports that the magnitude of the conditional noise prediction, $\|\tilde{\epsilon}_\theta(x_t, t, y)\|_2$, remains elevated within the basin and drops sharply upon exit. Leveraging this insight, we hypothesize that initial samples with smaller magnitudes $\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2$ are more likely to escape the basin earlier. To test this hypothesis, we introduce a novel method that adjusts x_T to obtain a modified sample \tilde{x}_T with reduced magnitude of conditional guidance $\|\tilde{\epsilon}_\theta(\tilde{x}_T, T, y)\|_2$. We then analyze the magnitude of the conditional noise predictions throughout the denoising process.

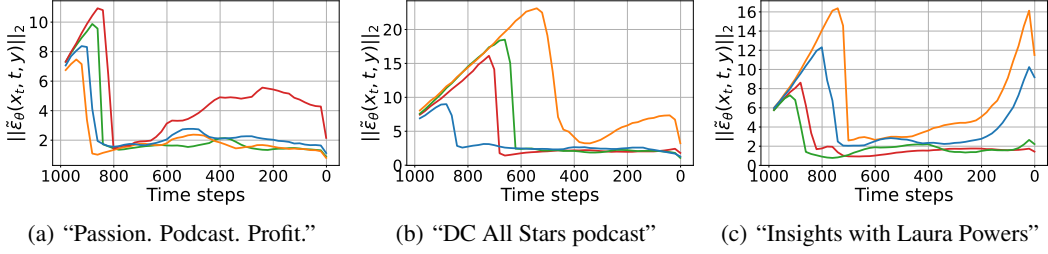


Figure 2: The magnitude of the conditional noise prediction at each timestep during sampling without CFG for three memorized prompts. Each line color corresponds to a different initial Gaussian sample. Transition points occur at different timesteps depending on the choice of initial sample.

3.1.1 Initial sample adjustment using sharpness

To facilitate clear analysis of the impact of $\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2$, we introduce a one-step adjustment method that modifies the initial sample x_T using a tunable step size that controls the strength of the magnitude reduction. To this end, we adopt the sharpness definition from Foret et al. [16] to reduce the magnitude of the conditional noise prediction. According to prior works [20, 11], the conditional noise prediction can be approximated in terms of the score function:

$$\tilde{\epsilon}_\theta(x_t, t, y) \approx -\sqrt{1 - \bar{\alpha}_t} \nabla_{x_t} \log p_\theta(y|x_t). \quad (4)$$

Letting $L(x_t) = -\log p_\theta(y|x_t)$, we obtain $\|\tilde{\epsilon}_\theta(x_t, t, y)\|_2 \approx \sqrt{1 - \bar{\alpha}_t} \|\nabla_{x_t} L(x_t)\|_2$, indicating that reducing the magnitude of conditional guidance corresponds to minimizing the gradient norm of $L(x_t)$. To achieve this, we adopt the following sharpness measure:

$$L_{\text{sharp}}(x_t) = \max_{\|\delta\|_2 \leq \rho} L(x_t + \delta) - L(x_t). \quad (5)$$

This definition captures the worst-case local increase in L within an ℓ_2 -ball of radius ρ centered at x_t , and penalizing this sharpness term has been shown to correspond to gradient norm minimization [16, 53]. Based on this formulation, we obtain the adjusted initial sample \tilde{x}_T , which yields the reduced magnitude of conditional guidance, as follows:

$$\tilde{x}_T = x_T - \gamma \nabla_{x_T} L_{\text{sharp}}(x_T), \quad (6)$$

where γ is a step size hyperparameter. To compute $\nabla_{x_T} L_{\text{sharp}}(x_T)$, we follow Foret et al. [16] and first approximate the inner maximization in Equation (5) using a first-order Taylor expansion to obtain

$$\delta^*(x_T) = \underset{\|\delta\|_2 \leq \rho}{\operatorname{argmax}} L(x_T + \delta) \approx \underset{\|\delta\|_2 \leq \rho}{\operatorname{argmax}} L(x_T) + \delta^\top \nabla_{x_T} L(x_T) = \underset{\|\delta\|_2 \leq \rho}{\operatorname{argmax}} \delta^\top \nabla_{x_T} L(x_T). \quad (7)$$

Since this approximation can be seen as a classical dual norm problem, its solution $\hat{\delta}(x_T)$ is given by:

$$\hat{\delta}(x_T) = \rho \cdot \frac{\nabla_{x_T} L(x_T)}{\|\nabla_{x_T} L(x_T)\|_2}. \quad (8)$$

By substituting $\hat{\delta}(x_T)$ from Equation (8) into the sharpness definition in Equation (5), and differentiating with respect to x_T , we obtain the update rule in Equation (6) as follows:

$$\begin{aligned} \tilde{x}_T &= x_T - \gamma \nabla_{x_T} (L(x_T + \hat{\delta}(x_T)) - L(x_T)) \\ &\approx x_T - \gamma (\nabla_{x_T} L(x_T)|_{x_T + \hat{\delta}(x_T)} - \nabla_{x_T} L(x_T)). \end{aligned} \quad (9)$$

Here, we follow Foret et al. [16] and approximate the gradient by omitting second-order terms. Substituting $\nabla_{x_T} L(x_T) \approx \frac{1}{\sqrt{1 - \bar{\alpha}_T}} \tilde{\epsilon}_\theta(x_T, T, y)$, we express the final adjustment rule in terms of the noise predictor:

$$\tilde{x}_T \approx x_T - \tilde{\gamma} (\tilde{\epsilon}_\theta(x_T + \hat{\delta}(x_T), T, y) - \tilde{\epsilon}_\theta(x_T, T, y)), \quad (10)$$

where $\hat{\delta}(x_T) = \rho \cdot \frac{\tilde{\epsilon}_\theta(x_T, T, y)}{\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2}$ and $\tilde{\gamma} = \frac{\gamma}{\sqrt{1 - \bar{\alpha}_T}}$. Both $\tilde{\gamma}$ and ρ are treated as hyperparameters.

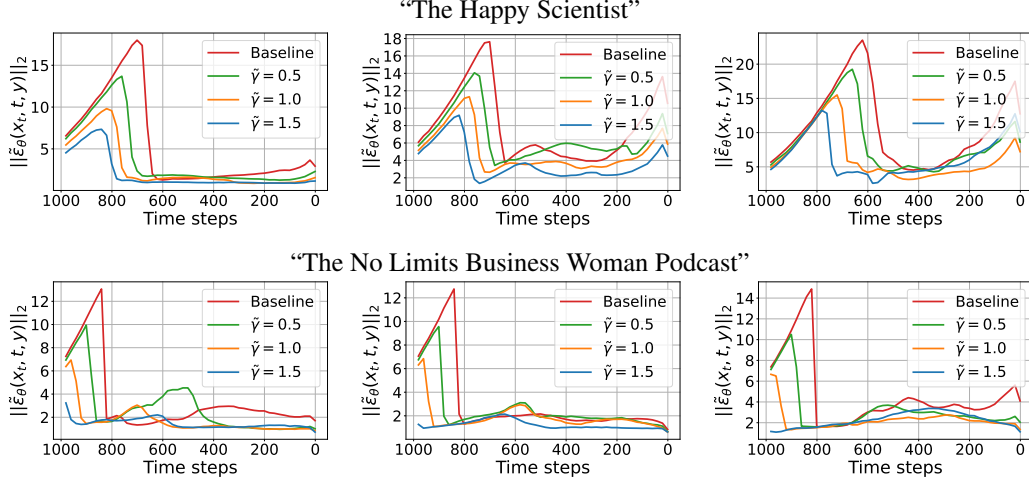


Figure 3: Plots showing the magnitude of the conditional noise prediction at each timestep during sampling without CFG under different adjustment strengths $\tilde{\gamma}$ for two memorized prompts. Each row corresponds to a different memorized prompt, and each column corresponds to a different initial Gaussian sample x_T . The baseline corresponds to $\tilde{\gamma} = 0$.

3.1.2 Magnitude analysis of conditional guidance for adjusted initial samples

In order to test our hypothesis, we use the adjusted initial sample \tilde{x}_T obtained via Equation (10) and observe the ℓ_2 -norm of the conditional noise prediction at each timestep when CFG is not applied. We employ Stable Diffusion [35] with $\rho = 50$ and vary the adjustment strength $\tilde{\gamma} \in \{0, 0.5, 1.0, 1.5\}$ to investigate the influence of the magnitude of conditional noise prediction at the initial timestep T . Figure 3 presents the results across different memorized prompts and initial samples. Each row corresponds to a different prompt, while each column shows the effect of applying the proposed adjustment to a different initial sample. The baseline denotes the unadjusted case ($\tilde{\gamma} = 0$). First, we observe that the magnitude of the conditional noise prediction at the initial timestep T decreases as $\tilde{\gamma}$ increases. This confirms that the proposed adjustment method successfully adjusts x_T to yield an updated sample \tilde{x}_T with reduced magnitude $\|\tilde{\epsilon}_\theta(\tilde{x}_T, T, y)\|_2$, as intended. Second—and more notably—we find that the transition point consistently occurs earlier in the denoising process as $\tilde{\gamma}$ increases. This trend holds across different prompts and initial samples, although the exact degree of the shift varies. These results support our hypothesis that initial samples with smaller magnitudes of conditional guidance $\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2$ are more likely to escape the attraction basin earlier, thereby advancing the transition point.

3.2 Proposed mitigation methods

Motivated by the above observations, we propose two mitigation approaches: Batch-wise mitigation and Per-sample mitigation. Batch-wise mitigation adjusts initial samples x_T collectively across a batch. This method reduces the magnitude of the conditional noise prediction using a small number of additional function evaluations (NFEs), producing adjusted samples \tilde{x}_T with minimal computational overhead. In contrast, Per-sample mitigation modifies each sample x_T individually. It reduces the magnitude of conditional noise prediction through direct backpropagation, continuing the update until the transition point is eliminated for each generation process. As a result, this method does not require an additional hyperparameter—a predefined timestep at which to begin applying CFG.

3.2.1 Batch-wise mitigation

For Batch-wise mitigation, we adopt the adjustment strategy introduced in the previous section. Given a batch of initial samples x_T , we apply the update rule in Equation (10) to the entire batch simultaneously. This update is repeated M times, where M is a tunable hyperparameter that determines the number of adjustment steps. The resulting adjusted samples \tilde{x}_T are then used as initial samples for the image generation process. Additionally, following Jain et al. [22], we apply

Algorithm 1: Batch-wise mitigation

Input: adjustment strength $\tilde{\gamma}$, sharpness parameter ρ , number of adjustments M , CFG application start timestep τ
Output: generated images x_0

Draw a batch of initial noise samples $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

/* Adjust the initial samples x_T with a small number of additional NFEs */

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

$$\begin{cases} \hat{\delta}(x_T) = \rho \cdot \frac{\tilde{\epsilon}_\theta(x_T, T, y)}{\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2} \\ \tilde{x}_T = x_T - \tilde{\gamma}(\tilde{\epsilon}_\theta(x_T + \hat{\delta}(x_T), T, y) - \tilde{\epsilon}_\theta(x_T, T, y)) \\ x_T = \tilde{x}_T \end{cases}$$

end

/* Perform the denoising process using the adjusted initial samples */

for $t = T$ **to** 1 **do**

$$\begin{cases} \text{if } t > \tau \text{ then} \\ \quad \epsilon^{\text{CFG}}(x_t, t, y) = \epsilon_\theta(x_t, t, y_{\text{null}}) \\ \text{else} \\ \quad \epsilon^{\text{CFG}}(x_t, t, y) = \epsilon_\theta(x_t, t, y_{\text{null}}) + w_{\text{CFG}}(\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, y_{\text{null}})) \\ \text{end} \\ x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}} \epsilon^{\text{CFG}}(x_t, t, y) \right) \end{cases}$$

end

CFG after a certain timestep to avoid generating memorized outputs. The complete procedure for Batch-wise mitigation is outlined in Algorithm 1.

3.2.2 Per-sample mitigation

For Per-sample mitigation, we adjust each initial sample x_T individually by directly minimizing $\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2$ via backpropagation. The adjustment continues until $\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2$ falls below a predefined target loss l_{target} . In contrast to Batch-wise mitigation, CFG is applied from the beginning of the sampling process. As shown in the rightmost plot of the bottom row in Figure 3, when the magnitude is sufficiently minimized (e.g., $\tilde{\gamma} = 1.5$), the transition point effectively disappears. This suggests that early application of CFG no longer induces memorization, thereby eliminating the need to predefine a specific timestep at which to begin applying CFG. Consequently, Per-sample mitigation enables the generation of non-memorized images even when CFG is applied from the initial timestep T . We conjecture that this behavior arises because sufficiently strong minimization yields an initial sample that lies outside the attraction basin. The pseudocode for Per-sample mitigation is provided in Appendix A.

4 Experiments

4.1 Experimental setup

Diffusion models and datasets. In line with prior works on memorization in diffusion models [40, 49, 34, 19, 8], we evaluate our proposed methods using Stable Diffusion v1.4 [35]. Although Stable Diffusion v2.0 exhibits fewer memorization issues due to being trained on a de-duplicated dataset, it still suffers from template memorization—a form of memorization in which a set of prompts leads to images that are highly similar to training samples, differing mainly in aspects such as color or style [34]. To account for this, we also employ Stable Diffusion v2.0 in our evaluations. For Stable Diffusion v1.4, we use 500 memorized prompts extracted from the LAION dataset [38], and for Stable Diffusion v2.0, we use 219 memorized prompts. These prompts are provided by Webster [47] and Ren et al. [34], respectively.

Evaluation metrics. We employ four complementary metrics to evaluate memorization, image-text alignment, image quality, and image diversity. To assess the degree of memorization in generated images, we use the Self-Supervised Copy Detection (SSCD) score [32], which measures object-level similarity between a generated image and its nearest neighbor in the training set. Image-text alignment is assessed with the CLIP score [33], which quantifies the semantic consistency between each generated image and its corresponding text prompt. For image quality and prompt alignment, we adopt ImageReward [50], while image diversity is evaluated using LPIPS [52]. Lower SSCD scores indicate stronger mitigation of memorization, whereas higher CLIP, LPIPS, and ImageReward scores reflect better performance. Detailed evaluations of image quality and diversity are presented in Appendix D.

Baselines. We compare our proposed methods against six baselines. The first is a no-mitigation baseline, where images are generated using Stable Diffusion without applying any memorization mitigation. Additionally, we consider five inference-time mitigation approaches: adding random tokens [40], adjusting the prompt embedding [49], scaling cross-attention logits [34], applying opposite guidance during the early stages of sampling [22], and applying guidance only after a specified timestep [22]. All baseline methods are applied at inference time and do not require access to the training data, making them suitable for fair comparison with our proposed approaches.

Implementation details. For each diffusion model, we generate 10 images per memorized prompt using identical inference configurations across all baselines and our proposed methods. Specifically, for Stable Diffusion v1.4, we employ a DDIM sampler [41] with 50 sampling steps and a CFG scale of 7, whereas for Stable Diffusion v2.0, we use a Euler discrete scheduler [26] with the same number of steps and CFG scale. Further implementation details are provided in Appendix B.

4.2 Impact of initial sample adjustment on memorization mitigation

To examine the effectiveness of our proposed initial sample adjustment in mitigating memorization while promoting image-text alignment, we compare mitigation results obtained with and without adjustment. The adjustment is implemented using the Batch-wise method. For the unadjusted case, the starting timestep for applying CFG, denoted as τ , is varied as $\tau \in \{780, 760, 740, 720\}$, while for the adjusted case it is set to $\tau \in \{900, 880, 860, 840\}$. This design ensures that we can isolate the impact of the adjustment itself, given that τ influences mitigation performance. The results are illustrated in Figure 4. On the No adjustment curve, the rightmost and leftmost data points correspond to $\tau = 780$ and $\tau = 720$, respectively; for the With adjustment curve, they correspond to $\tau = 900$ and $\tau = 840$. Despite the earlier application of CFG (*i.e.*, higher τ), the adjusted cases consistently achieve lower SSCD scores than their unadjusted counterparts at comparable CLIP scores, indicating that the proposed adjustment effectively shifts the transition point earlier in the denoising process and enables memorization mitigation even when CFG is applied early. More importantly, the adjustment yields a more favorable trade-off between SSCD score and CLIP score. These results demonstrate that our initial sample adjustment effectively reduces memorization while encouraging image-text alignment.

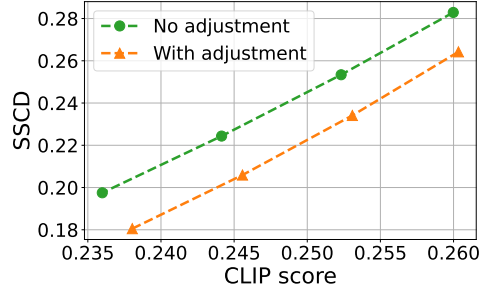


Figure 4: Comparison of SSCD and CLIP scores with and without initial sample adjustment.

4.3 Comparison with baselines

In this subsection, we compare our proposed methods with baseline approaches by analyzing how SSCD and CLIP scores vary across different mitigation strengths, following prior works [34, 49]. We evaluate both the baselines and our Per-sample method under five distinct hyperparameter settings. It is worth noting that mitigation strength in our methods is closely tied to the value of $\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2$. To control this value, the Batch-wise method employs three hyperparameters ($\tilde{\gamma}$, ρ , and M), whereas the Per-sample method uses only one (l_{target}). Since both approaches share the core idea that adjusting the initial noise x_T to minimize $\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2$ improves mitigation performance, we report the performance of Per-sample across multiple l_{target} values. Detailed hyperparameter configurations

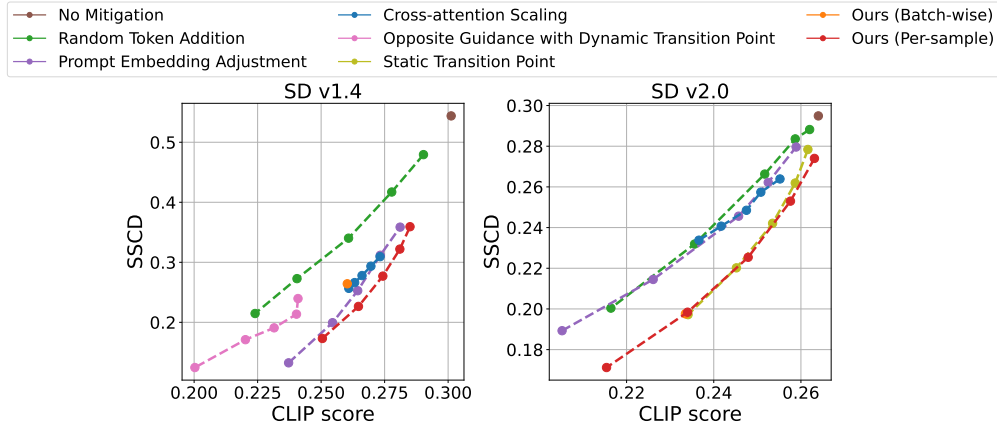


Figure 5: Comparison of SSSD and CLIP scores among different mitigation methods under Stable Diffusion v1.4 and v2.0. Lower SSSD scores indicate stronger memorization mitigation, while higher CLIP scores indicate better image-text alignment.

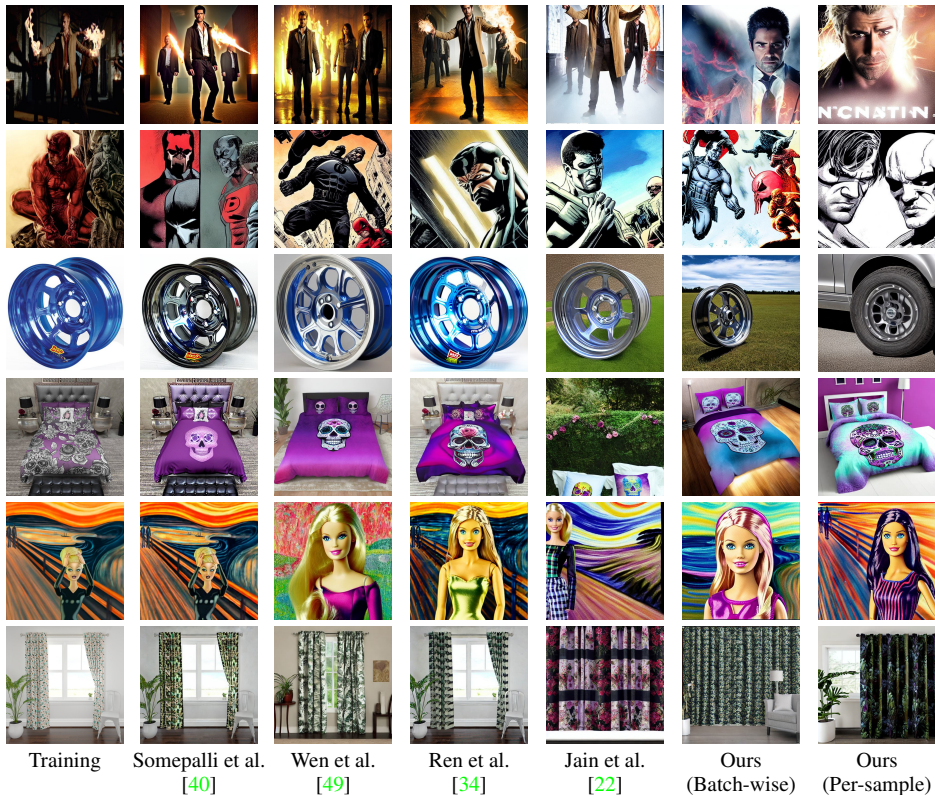


Figure 6: Qualitative comparison of memorization mitigation results. Each column shows generations produced by various baseline methods and our proposed approaches. The prompts used for the generations are provided in Appendix G.

for all methods are provided in Appendix B. The results in Figure 5 show that Batch-wise achieves performance comparable to Cross-attention Scaling under Stable Diffusion v1.4, while achieving the best performance under v2.0. More importantly, Per-sample consistently provides the most favorable SSSD–CLIP trade-off across both diffusion models, highlighting its superior effectiveness in mitigating memorization. Qualitative comparisons are presented in Figure 6.

A performance gap is observed between the Batch-wise and Per-sample methods under Stable Diffusion v1.4. We conjecture that this difference arises because the optimal hyperparameters for

the Batch-wise method vary across prompts and initial samples. As shown in Figure 3, a fixed set of hyperparameters can lead to different degrees of magnitude reduction in $\tilde{\epsilon}_\theta(x_T, T, y)$, resulting in varying transition points depending on the prompt and the initial sample. Moreover, since CFG must be applied after—but not far beyond—the transition point, the optimal timestep for CFG application is inherently sample-specific in the Batch-wise setting. In contrast, the Per-sample method directly minimizes the magnitude of $\tilde{\epsilon}_\theta(x_T, T, y)$ via backpropagation with fixed hyperparameters, effectively eliminating the transition point for each individual generation. This enables robust memorization mitigation while improving image-text alignment, as demonstrated in Figure 5. Although the Batch-wise approach is less adaptive and slightly underperforms relative to the Per-sample method, it still achieves competitive performance.

5 Related work

Memorization in diffusion models. Beyond supervised models [4] and language models [3], diffusion models [5, 39, 40] have also been found to exhibit a memorization problem, indicating that models tend to memorize and regenerate training data. This phenomenon raises significant privacy concerns, as it can result in unintended data leakage. To address this issue, various mitigation strategies have been explored. While early approaches sought to mitigate memorization either during training or at inference, training-time mitigation typically requires access to the original dataset and substantial computational resources, limiting its practicality. Consequently, recent efforts have shifted toward inference-time mitigation strategies [7, 22, 19, 8, 34]. For example, Wen et al. [49] proposes adjusting prompt embeddings to minimize the magnitude of the conditional noise prediction, while Chen et al. [8] introduces attention masking to enhance this adjustment. Somepalli et al. [40] mitigates memorization by perturbing prompts through token addition or replacement, and Jain et al. [22] suggests applying opposite guidance during early denoising steps to counteract memorization induced by CFG. Our work follows this line of research, proposing inference-time mitigation strategies that adjust initial noise samples to reduce memorization.

Impact of initial noise samples. Recent studies have increasingly focused on the impact of initial noise samples on image generation in diffusion models, exploring how modifying these samples can address various challenges, such as improving image-text alignment and enhancing image quality [51, 31, 17, 44, 46, 6, 37, 9]. For example, Wallace et al. [45] proposes an end-to-end framework that backpropagates through the entire diffusion process to adjust the initial noise sample, thereby improving classifier guidance. Guo et al. [18] utilizes attention maps to guide the refinement of initial noise samples, aiming to mitigate semantic errors such as subject neglect, subject mixing, and incorrect binding. Similarly, Eyring et al. [15] addresses poor semantic fidelity in text-to-image models by optimizing the initial noise samples based on reward feedback.

Sharpness aware minimization. The sharpness term used to minimize the magnitude of the conditional noise prediction in our method was proposed in Sharpness-Aware Minimization (SAM) [16]. SAM was originally designed to seek flatter minima in the loss landscape, motivated by several studies showing that flatter minima are associated with better model generalization [27, 25, 14]. By simultaneously minimizing both the loss value and the sharpness of the loss surface, SAM aims to find model parameters that reside in regions where the loss values remain consistently low. SAM has achieved state-of-the-art generalization performance, resulting in numerous follow-up works that aim to improve it by introducing surrogate loss function [54], proposing geometric measures of neighborhood [29, 28], or enhancing training efficiency [12, 13, 24]. However, a deep understanding of which components of SAM drive its generalization improvement remains limited. As a result, there have been significant efforts to analyze the underlying mechanisms of SAM [1, 48, 10, 53]. For instance, Zhao et al. [53] reveals that SAM effectively minimizes the gradient norm during optimization.

In contrast to previous works, to the best of our knowledge, this paper is the first to investigate the impact of the initial noise sample on mitigating memorization in diffusion models. By introducing a novel method that adjusts the initial sample by minimizing the magnitude of the conditional noise prediction, we analyze the relationship between the magnitude of conditional guidance and the transition point. Building on this analysis, we propose two novel mitigation strategies that effectively reduce memorization while promoting image-text alignment.

6 Limitations

While our proposed methods demonstrate strong effectiveness, they also have certain limitations. The Per-sample method introduces a slight increase in computational overhead, and the performance of the Batch-wise method can vary with hyperparameter selection. However, we believe that the additional cost of the Per-sample method is well justified by its effectiveness in mitigating memorization and enhancing privacy and intellectual property protection. Moreover, the close relationship between the hyperparameters of the Batch-wise method and the goal of reducing the magnitude of conditional guidance offers a clear and efficient direction for hyperparameter tuning.

7 Conclusion

We present a novel perspective on the role of initial samples in mitigating memorization in diffusion models, viewed through the lens of the attraction basin. We observe that different initializations result in different transition points. Given that earlier transition points lead to non-memorized images that remain well-aligned with text prompts, we further analyze which initial samples facilitate faster escape from the attraction basin and find that adjusting initial noise samples to achieve a smaller magnitude of the conditional noise prediction results in earlier transition points. Building on these insights, we propose two novel inference-time mitigation strategies—Batch-wise and Per-sample approaches—which adjust initial samples collectively or individually and use them for image generation. Experimental results validate our findings and demonstrate the effectiveness of the proposed strategies in mitigating memorization while improving image-text alignment.

Acknowledgements

This work is in part supported by the National Research Foundation of Korea (NRF, RS-2024-00451435(20%), RS-2024-00413957(20%)), Institute of Information & communications Technology Planning & Evaluation (IITP, RS-2021-II212068(10%), RS-2025-02305453(10%), RS-2025-02273157(10%), RS-2025-25442149(10%), 2021-0-00180(10%), RS-2021-II211343(10%)) grant funded by the Ministry of Science and ICT (MSIT), Institute of New Media and Communications(INMAC), and the BK21 FOUR program of the Education, Artificial Intelligence Graduate School Program (Seoul National University), and Research Program for Future ICT Pioneers, Seoul National University in 2025.

References

- [1] Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International conference on machine learning*, pages 639–668. PMLR, 2022.
- [2] Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. In *International Conference on Machine Learning*, pages 3462–3483. PMLR, 2024.
- [3] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.
- [4] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.
- [5] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.
- [6] Changgu Chen, Libing Yang, Xiaoyan Yang, Lianggangxu Chen, Gaoqi He, Changbo Wang, and Yang Li. Find: Fine-tuning initial noise distribution with policy optimization for diffusion models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6735–6744, 2024.

- [7] Chen Chen, Daochang Liu, and Chang Xu. Towards memorization-free diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8425–8434, 2024.
- [8] Chen Chen, Daochang Liu, Mubarak Shah, and Chang Xu. Exploring local memorization in diffusion models via bright ending attention. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=p4cLtzk4oe>.
- [9] Sherry X Chen, Yaron Vaxman, Elad Ben Baruch, David Asulin, Aviad Moreshet, Kuo-Chin Lien, Misha Sra, and Pradeep Sen. Tino-edit: Timestep and noise optimization for robust diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6337–6346, 2024.
- [10] Zixiang Chen, Junkai Zhang, Yiwen Kou, Xiangning Chen, Cho-Jui Hsieh, and Quanquan Gu. Why does sharpness-aware minimization generalize better than sgd? *Advances in neural information processing systems*, 36:72325–72376, 2023.
- [11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [12] Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Tan. Efficient sharpness-aware minimization for improved training of neural networks. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=n00eTdNRG0Q>.
- [13] Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. *Advances in Neural Information Processing Systems*, 35:23439–23451, 2022.
- [14] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.
- [15] Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. Reno: Enhancing one-step text-to-image models through reward-based noise optimization. *Advances in Neural Information Processing Systems*, 37:125487–125519, 2024.
- [16] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=6Tm1mposlrm>.
- [17] Paul Grimal, Hervé Le Borgne, Olivier Ferret, and Julien Tourille. Tiam-a metric for evaluating alignment in text-to-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2890–2899, 2024.
- [18] Xiefan Guo, Jinlin Liu, Miaomiao Cui, Jiankai Li, Hongyu Yang, and Di Huang. Initno: Boosting text-to-image diffusion models via initial noise optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9380–9389, 2024.
- [19] Dominik Hintersdorf, Lukas Struppek, Kristian Kersting, Adam Dziedzic, and Franziska Boenisch. Finding nemo: Localizing neurons responsible for memorization in diffusion models. *Advances in Neural Information Processing Systems*, 37:88236–88278, 2024.
- [20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [22] Anubhav Jain, Yuya Kobayashi, Takashi Shibuya, Yuhta Takida, Nasir Memon, Julian Togelius, and Yuki Mitsufuji. Classifier-free guidance inside the attraction basin may cause memorization. *arXiv preprint arXiv:2411.16738*, 2024.

- [23] Harry H Jiang, Lauren Brown, Jessica Cheng, Mehtab Khan, Abhishek Gupta, Deja Workman, Alex Hanna, Johnathan Flowers, and Timnit Gebru. Ai art and its impact on artists. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 363–374, 2023.
- [24] Weisen Jiang, Hansi Yang, Yu Zhang, and James Kwok. An adaptive policy to employ sharpness-aware minimization. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=6wL7-M2BC->.
- [25] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.
- [26] 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.
- [27] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.
- [28] Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher sam: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pages 11148–11161. PMLR, 2022.
- [29] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021.
- [30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [31] Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided image synthesis via initial image editing in diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5321–5329, 2023.
- [32] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14532–14542, 2022.
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.
- [34] Jie Ren, Yaxin Li, Shenglai Zeng, Han Xu, Lingjuan Lyu, Yue Xing, and Jiliang Tang. Unveiling and mitigating memorization in text-to-image diffusion models through cross attention. In *European Conference on Computer Vision*, pages 340–356. Springer, 2024.
- [35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [36] Dvir Samuel, Rami Ben-Ari, Nir Darshan, Haggai Maron, and Gal Chechik. Norm-guided latent space exploration for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:57863–57875, 2023.
- [37] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4695–4703, 2024.

- [38] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.
- [39] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6048–6058, 2023.
- [40] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023.
- [41] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=St1giarCHLP>.
- [42] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.
- [43] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=PXTIG12RRHS>.
- [44] Wenqiang Sun, Teng Li, Zehong Lin, and Jun Zhang. Spatial-aware latent initialization for controllable image generation. *arXiv preprint arXiv:2401.16157*, 2024.
- [45] Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7280–7290, 2023.
- [46] Ruoyu Wang, Huayang Huang, Ye Zhu, Olga Russakovsky, and Yu Wu. The silent prompt: Initial noise as implicit guidance for goal-driven image generation. *arXiv preprint arXiv:2412.05101*, 2024.
- [47] Ryan Webster. A reproducible extraction of training images from diffusion models, 2023. URL <https://arxiv.org/abs/2305.08694>.
- [48] Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=5spDgWmpY6x>.
- [49] 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*, 2024. URL <https://openreview.net/forum?id=84n3UwkH7b>.
- [50] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.
- [51] Katherine Xu, Lingzhi Zhang, and Jianbo Shi. Good seed makes a good crop: Discovering secret seeds in text-to-image diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3024–3034. IEEE, 2025.
- [52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [53] Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International conference on machine learning*, pages 26982–26992. PMLR, 2022.

- [54] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, sekhar tatikonda, James s Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=edONMAhLu->.

NeurIPS Paper Checklist

1. Claims

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

Answer: [Yes]

Justification: We explicitly outline the contributions of our paper in both the abstract and introduction sections. Specifically, our contributions are itemized in the final paragraph of the Introduction section.

Guidelines:

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

2. Limitations

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

Answer: [Yes]

Justification: We elucidate the limitations of our method in the Limitations section.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [NA]

Justification: This paper does not present any theoretical results; no theorems or formal assumptions are introduced.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: All experimental details are outlined in Section 4 (Experiments) and further elaborated in Appendix A (Implementation details) and Appendix B (Pseudocode for Per-sample mitigation).

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes]

Justification: We adhere to the NeurIPS code submission policy.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: Experimental details are outlined in Section 4 (Experiments) and Appendix A.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [Yes]

Justification: We follow the evaluation protocols established by existing works in this area. To highlight our results, we report SSCD, CLIP score, and FID metrics.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

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

Answer: [Yes]

Justification: Information regarding computer resources is provided in Appendix A.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: We fully comply with the code of ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [Yes]

Justification: We highlight data leakage issues caused by memorization in text-to-image diffusion models in the Introduction and discuss how our proposed methods address these issues in the Conclusion.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. Safeguards

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

Answer: [NA]

Justification: Our research method and area are unrelated to such issues.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: We illustrate the sources of assets in Section 4 (Experiments) and Appendix A (Implementation details).

Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [NA]

Justification: We do not release new assets.

Guidelines:

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

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

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

Answer: [NA]

Justification: We did not conduct experiments with human subjects.

Guidelines:

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

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

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

Answer: [NA]

Justification: We did not conduct experiments with human subjects.

Guidelines:

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

16. Declaration of LLM usage

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

Answer: [NA]

Justification: The proposed method was developed without the involvement of any LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A Pseudocode for Per-sample mitigation

We present the pseudocode for the Per-sample mitigation method in Figure 7.

```
1 def adj_latent(latent, t, prompt_embeds, lr, target_loss):
2     latent.requires_grad = True
3     optim = AdamW([latent], lr=lr)
4     while True:
5         pred_uncond, pred_text = unet(latent, t, prompt_embeds)
6         loss = norm(pred_text - pred_uncond)
7         if loss < target_loss:
8             break
9         optim.zero_grad(); loss.backward(); optim.step()
10    return latent.detach()
11
12 def per_sample_mitigation(prompt_embeds, cfg_scale, lr, target_loss):
13    latent = randn(latent_shape)
14    for i, t in enumerate(timesteps):
15        # Adjust the initial sample using backpropagation
16        if i == 0:
17            latent = adj_latent(latent, t, prompt_embeds, lr, target_loss)
18        # Perform the denoising process with the adjusted initial sample
19        pred_uncond, pred_text = unet(latent, t, prompt_embeds)
20        noise_pred = pred_uncond + cfg_scale * (pred_text - pred_uncond)
21        latent = scheduler.step(noise_pred, t, latent)
22    return latent
```

Figure 7: Pseudocode illustrating the Per-sample mitigation method.

B Implementation details

All experiments were conducted using an NVIDIA A100 GPU. The implementation environment included Hugging Face 0.23.4, CUDA 12.0, Python 3.8.0, and PyTorch 2.3.1 with Torchvision 0.18.1. Detailed configurations for the proposed mitigation strategies and baselines are outlined below.

Batch-wise mitigation. For Stable Diffusion v1.4, the adjustment strength is set to $\tilde{\gamma} = 0.7$, the sharpness parameter to $\rho = 50.0$, and the number of adjustments to $M = 2$. CFG is applied with a guidance scale of 7.0 for timesteps $t \leq \tau$, and disabled (*i.e.*, set to 0.0) for $t > \tau$, where the CFG application start timestep is set to $\tau = 900$. For Stable Diffusion v2.0, we set $\tilde{\gamma} = 0.7$, $\rho = 30.0$, $M = 4$, and $\tau = 917.4490$.

Per-sample mitigation. For Stable Diffusion v1.4, the initial noise sample is adjusted using the AdamW optimizer [30] with a learning rate of 0.01, while all other hyperparameters remain at their default values. The target threshold is varied as $l_{\text{target}} \in \{0.7, 0.9, 1.1, 1.3, 1.5\}$. For Stable Diffusion v2.0, the initial noise sample is adjusted using the same optimizer with a learning rate of 0.1, and the target threshold is varied as $l_{\text{target}} \in \{5, 8, 10, 15, 20\}$.

Baselines. For Random Token Addition, the number of added tokens is set to $\{1, 2, 4, 6, 8\}$ for both diffusion models. For Prompt Embedding Adjustment, the target loss is set to $l_{\text{target}} \in \{3, 4, 5, 6, 7\}$ for Stable Diffusion v1.4 and $l_{\text{target}} \in \{50, 60, 70, 80, 90\}$ for Stable Diffusion v2.0. For Cross-attention Scaling, the re-scaling factor is set to $C \in \{1.15, 1.2, 1.25, 1.3, 1.35\}$ for both diffusion models. For Opposite Guidance with Dynamic Transition Point, the opposite guidance scale is

set to $\{1, 3, 5, 7, 9\}$. For Static Transition Point, the CFG application start timestep is set to $\tau \in \{999.0, 978.6122, 958.2245, 937.8367, 917.4490\}$.

C Analysis of distributional shifts induced by initial noise adjustment

We observed that using initial noise with weaker conditional guidance leads to an earlier transition point, thereby improving memorization mitigation performance. Consequently, our objective is to effectively reduce the norm $\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2$. A natural question arises: is it appropriate to minimize this norm without an explicit regularization term to prevent the adjusted initial noise from becoming out-of-distribution (OOD)? We argue that the weight decay in the AdamW optimizer partially fulfills this role. Furthermore, with the default weight decay value of $w = 0.01$, we observed no degradation in either mitigation performance or image quality. The following subsections provide both theoretical and empirical evidence supporting this claim.

C.1 Effect of weight decay in mitigating distributional shift

According to prior works [2, 36], the L2 norm of the initial noise $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ follows a chi distribution:

$$\|x_T\|_2 = \sqrt{\sum_{i=1}^d x_T^i{}^2} \sim \chi^d = \|x_T\|_2^{d-1} e^{-\|x_T\|_2^2/2} / (2^{d/2-1} \Gamma(\frac{d}{2})), \quad (11)$$

where d denotes the dimensionality of x_T and $\Gamma(\cdot)$ is the Gamma function. Based on this, to prevent the adjusted initial noise from deviating into an OOD state, one could consider minimizing the objective

$$\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2 - w \log p(\|x_T\|_2), \quad (12)$$

where p represents the chi distribution and w is a weighting coefficient. By substituting the chi distribution from Equation (11) into $p(\|x_T\|_2)$ in Equation (12), the objective can be rewritten as

$$\|\tilde{\epsilon}_\theta(x_T, T, y)\|_2 + c - w(d-1) \log \|x_T\|_2 + w \frac{\|x_T\|_2^2}{2}, \quad (13)$$

where c is a constant independent of x_T . We argue that the weight decay term in the AdamW optimizer implicitly minimizes the final quadratic term, $\frac{\|x_T\|_2^2}{2}$, thereby constraining adjusted initial samples from becoming OOD. To empirically validate this, we compared 5000 unadjusted and 5000 adjusted initial noise samples, measuring the Jensen–Shannon divergence (JSD) between their distributions under varying weight decay values. The results shown in Table 1 demonstrate that as w increases, JSD decreases. This suggests that stronger weight decay encourages the adjusted samples to remain closer to the original noise distribution.

Table 1: Jensen–Shannon divergence between adjusted and unadjusted initial noise distributions under varying weight decay values.

	$w = 0.01$	$w = 0.05$	$w = 0.1$	$w = 0.2$
JSD	0.0665	0.0663	0.0638	0.046

C.2 Effect of weight decay on mitigation performance

We varied the weight decay parameter w as described in the previous subsection and evaluated the corresponding performance. The results are presented in Table 2. In this experiment, the target threshold was fixed at $l_{\text{target}} = 0.9$. Across all tested settings, the performance remained largely consistent regardless of the value of w , suggesting that the default weight decay value $w = 0.01$ does not cause degradation in either mitigation performance or image quality. We attribute this stability to the fact that the default value does not cause the adjusted initial noise to deviate significantly from the original distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This interpretation is further supported by the JSD results in Table 1, where a JSD value of 0.0665 at $w = 0.01$ indicates minimal distributional shift.

Table 2: Performance comparison across different weight decay values.

	$w = 0.01$	$w = 0.05$	$w = 0.1$	$w = 0.2$
SSCD	0.2265	0.2265	0.2265	0.2275
CLIP	0.2647	0.2647	0.2647	0.2657
LPIPS	0.7615	0.7615	0.7615	0.7610
ImageReward	-0.3522	-0.3522	-0.3522	-0.3501

D Performance comparison on image quality and diversity

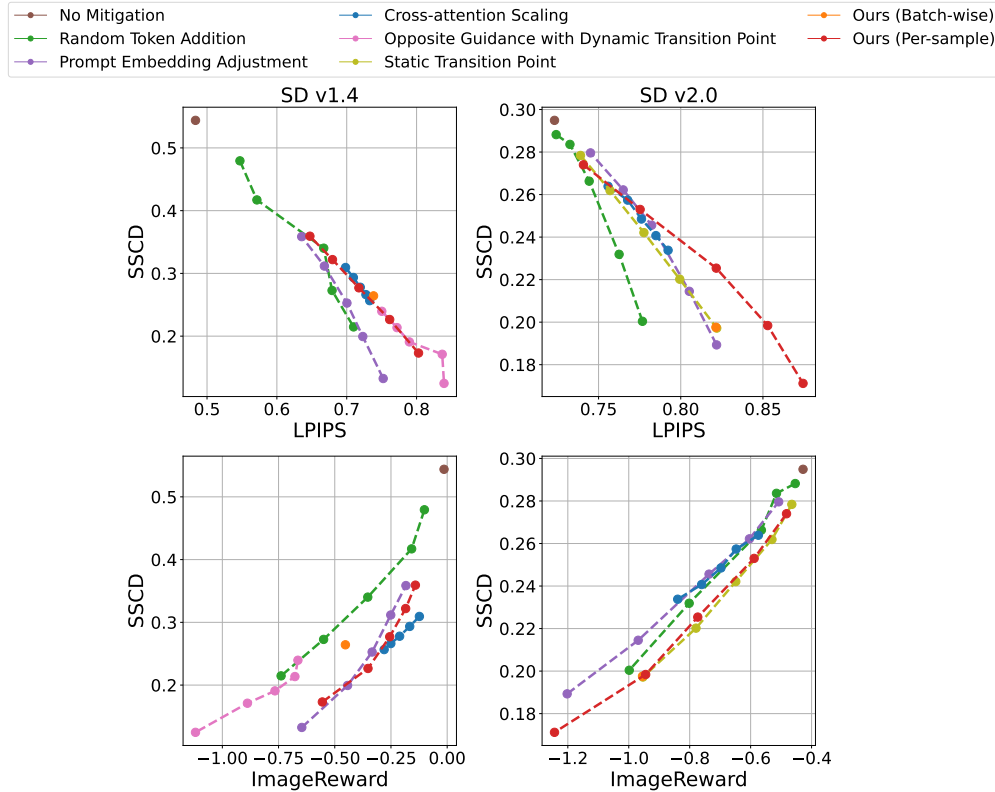


Figure 8: Comparison of different mitigation methods under Stable Diffusion v1.4 and v2.0. The first row shows SSCD versus LPIPS, while the second row shows SSCD versus ImageReward. Lower SSCD scores indicate stronger memorization mitigation, whereas higher LPIPS and ImageReward values reflect greater image diversity and quality.

We compare our proposed methods with baseline approaches by analyzing how LPIPS and ImageReward values vary across different mitigation strengths. To this end, we present two sets of results: SSCD versus LPIPS and SSCD versus ImageReward, using the same hyperparameters as in Figure 5. The results are shown in Figure 8. Since effective memorization mitigation enables the generation of non-memorized images, it also promotes greater diversity in the outputs, as illustrated in the SSCD–LPIPS results in the first row. Under Stable Diffusion v1.4, the Batch-wise method achieves the highest LPIPS values at comparable SSCD scores, while Per-sample achieves the second-highest LPIPS values at lower SSCD scores. In contrast, under Stable Diffusion v2.0, Batch-wise yields the second-highest LPIPS values, whereas Per-sample attains the highest LPIPS values with a substantial margin. The SSCD–ImageReward results in the second row further show that the Per-sample approach provides the most favorable trade-off between memorization mitigation and image quality across both diffusion models. These findings demonstrate that our proposed methods introduce minimal quality degradation while substantially enhancing image diversity, thereby improving the practical utility of diffusion models.

E Magnitude of conditional noise prediction across varying adjustment strengths

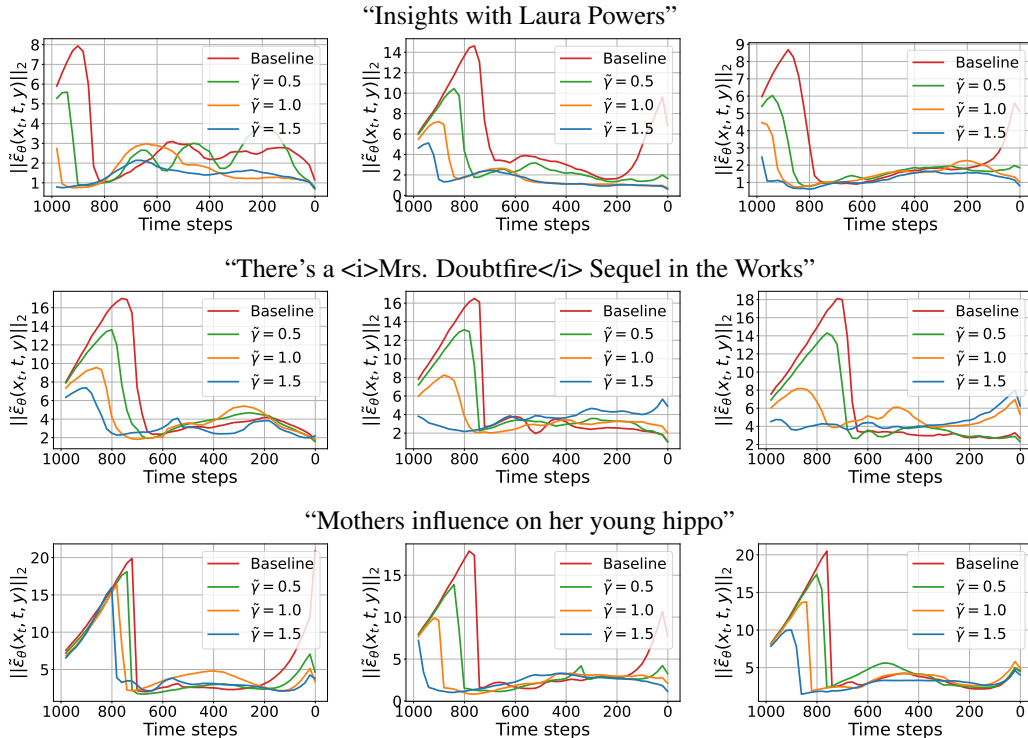


Figure 9: Magnitude of $\tilde{\epsilon}_\theta(x_t, t, y)$ at each timestep during sampling without CFG under varying adjustment strengths $\tilde{\gamma}$. Each row corresponds to a different memorized prompt, and each column corresponds to a distinct initial noise sample.

We provide additional plots in Figure 9 illustrating the magnitude of conditional guidance at each timestep during denoising without CFG, under varying adjustment strengths $\tilde{\gamma}$. The results show a consistent trend: as $\tilde{\gamma}$ increases, the magnitude of $\tilde{\epsilon}_\theta(\tilde{x}_T, T, y)$ decreases, leading to an earlier occurrence of the transition point across different prompts and initial noise samples.

F Magnitude analysis of conditional guidance under Per-sample mitigation

In the Per-sample method, we adjust the initial noise sample x_T by minimizing the magnitude of $\tilde{\epsilon}_\theta(x_T, T, y)$ until the transition point effectively disappears. To validate this effect, we compare the magnitude of the conditional noise prediction with and without applying the Per-sample method. In these experiments, we set the target loss $l_{\text{target}} = 0.9$ and use a learning rate of 0.01. Figure 10 presents the results: each row corresponds to a different memorized prompt, and each column corresponds to a different initial noise sample x_T . The Per-sample method significantly reduces the magnitude of $\tilde{\epsilon}_\theta(x_T, T, y)$. Moreover, unlike the baseline, no sharp drop in magnitude is observed, indicating that the transition point has been effectively eliminated. This enables CFG to be applied from the beginning of the denoising process.

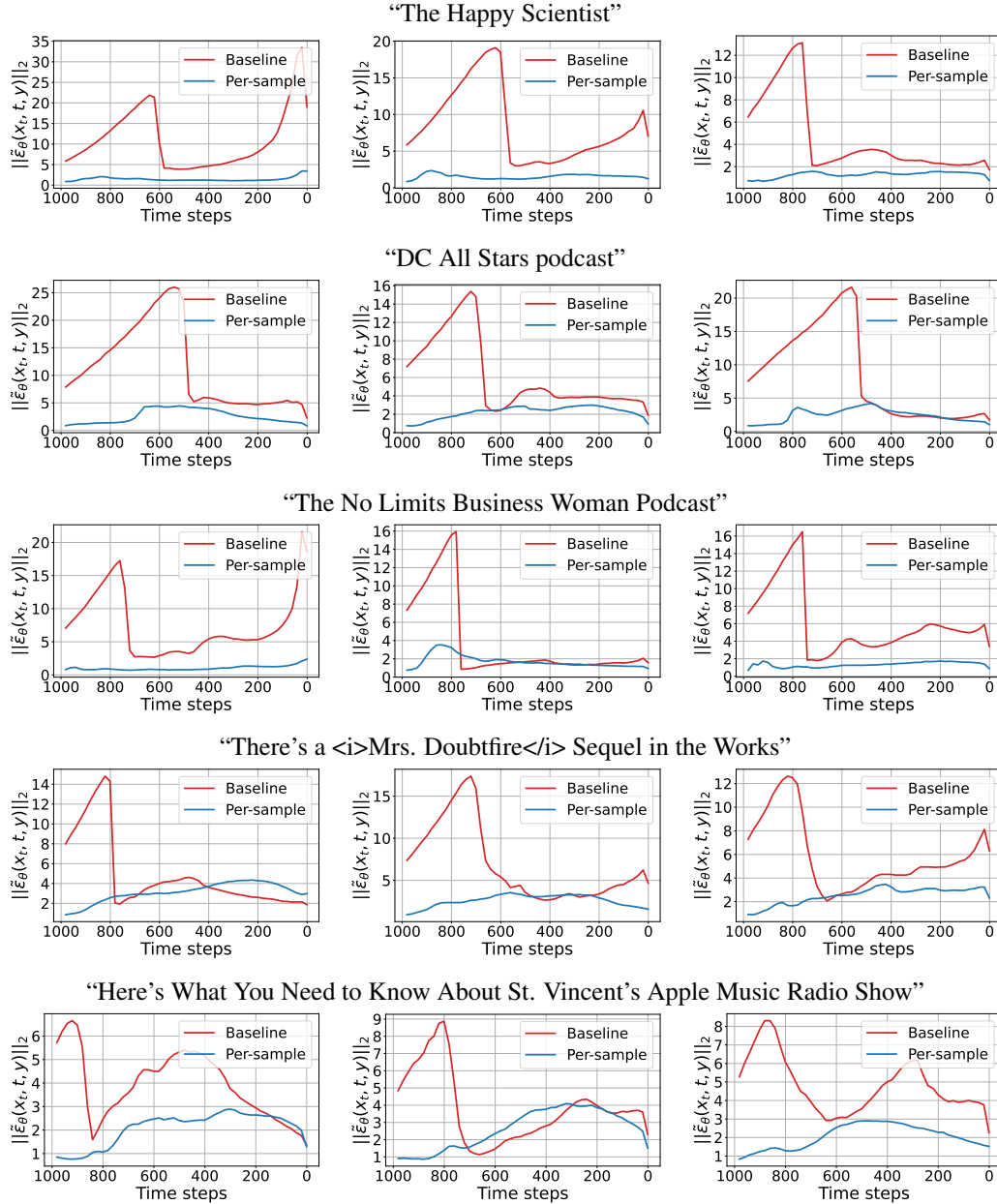


Figure 10: The magnitude of the conditional noise prediction at each timestep during sampling without CFG for five memorized prompts. The baseline corresponds to the case without adjustment, while the Per-sample method is applied with a target loss $l_{\text{target}} = 0.9$.

G Prompts for Figure 6

The following memorized prompts were used to generate the samples shown in Figure 6:

- Watch the Trailer for NBC's *Constantine*
- As Punisher Joins *Daredevil* Season Two, Who Will the New Villain Be?
- Aero 50-975035BLU 50 Series 15x7 Inch Wheel, 5 on 5 Inch BP 3-1/2 BS
- Signature Purple Ombre Sugar Skull and Rose Bedding
- If Barbie Were The Face of The World's Most Famous Paintings
- Plymouth Curtain Panel featuring Madelyn - White Botanical Floral Large Scale by heatherdutton

H Additional qualitative results

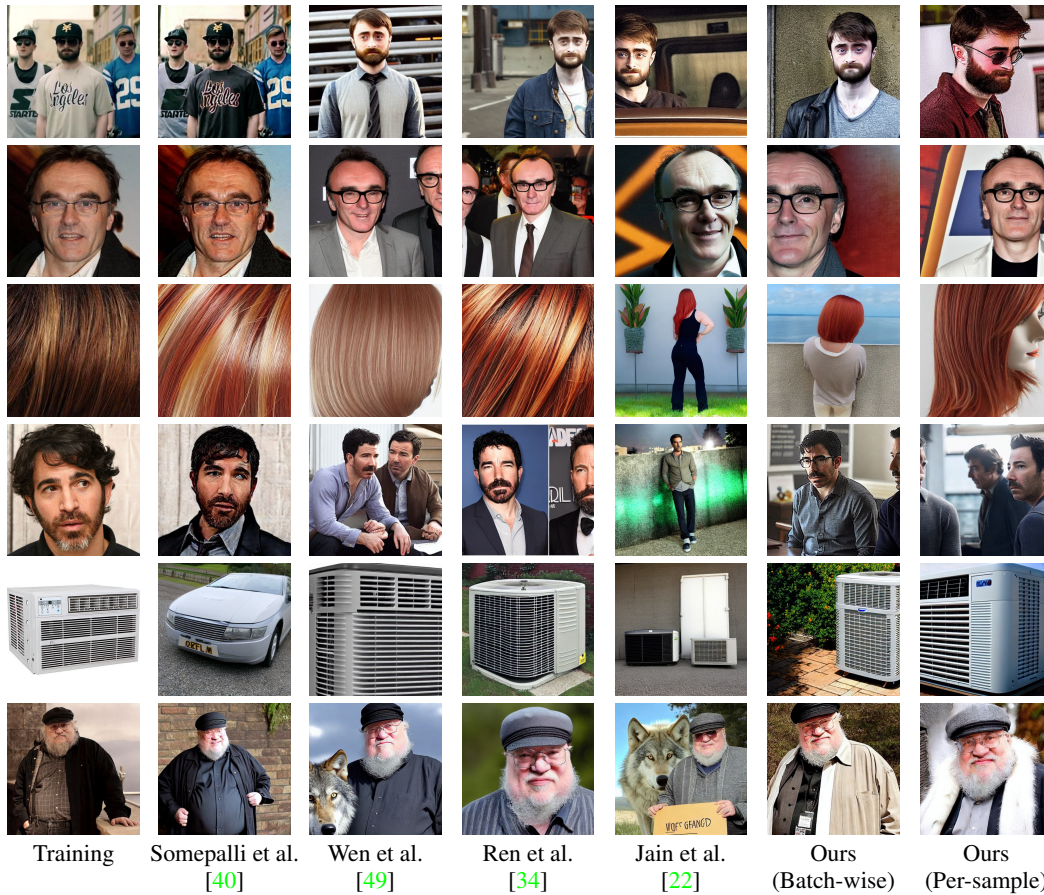


Figure 11: Additional qualitative comparison between our proposed methods and baseline approaches. Each column presents images produced by a baseline or one of our methods, with the leftmost column showing the corresponding training images for reference.

We present additional qualitative results comparing the baseline methods with our proposed approaches in Figure 11. The memorized prompts used to generate these samples are listed below:

- Daniel Radcliffe Dons a Beard and Saggy Jeans in Trailer for BBC GTA Miniseries <i>The Gamechangers</i>
- Director Danny Boyle Is Headed To TV With FX Deal
- 30A27S4 - Shaded Peach - Brown Red/Strawberry Blonde Blend, Shaded w/ Dk Brown
- Chris Messina In Talks to Star Alongside Ben Affleck in <i>Live By Night</i>
- Air Conditioners & Parts
- George R.R. Martin Donates \$10,000 to Wolf Sanctuary for a 13-Year-Old Fan

I Computational cost analysis

Table 3: Computational cost of different mitigation methods.

	gpu memory usage (MB)	inference time (sec)
No Mitigation	3564.94	1.85
Random Token Addition	3573.57	1.85
Prompt Embedding Adjustment	8115.53	1.98
Cross-attention Scaling	3170.38	1.93
Opposite Guidance with Dynamic Transition Point	3573.57	1.87
Ours (Batch-wise)	3573.57	2.01
Ours (Per-sample)	9701.62	2.8

We conducted an additional experiment to compare baseline mitigation methods with our proposed approaches in terms of computational cost. For a fair comparison, we measured GPU memory consumption and inference time per image. The results, presented in Table 3, show that the Batch-wise method introduces negligible computational overhead, while the Per-sample method requires relatively higher memory usage and longer inference time. However, as demonstrated in Figure 5, the Per-sample method consistently achieves superior memorization mitigation and robustness across different models. Given that effective mitigation is crucial for enhancing privacy and safeguarding intellectual property while reducing potential risks, we believe that the additional computational cost of the Per-sample approach is a reasonable trade-off.