MEMORIZATION DETECTION IN DIFFUSION MODELS VIA TEXT EMBEDDING INTERPOLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image diffusion models generate high-fidelity images from a text prompt but exhibit memorization on certain prompts where the model reproduces a training image. A widely adopted approach to memorization detection quantifies the difference between the conditional and unconditional scores. We express the magnitude of this difference as the norm of the integral of the conditional score's derivative along the linear interpolation from the unconditional embedding to the prompt embedding. For memorized prompts with an anomalously large score difference, the rate of change of the conditional score remains relatively stable over most of the interpolation path and rises sharply within a narrow interval near the prompt embedding. The generated image follows the same trajectory, remaining aligned with the prompt semantics across the smooth segment and switching to the memorized training image across the narrow interval. We measure this sharp rise through the condition Jacobian evaluated at the prompt embedding, which yields a detection performance that achieves state-of-the-art accuracy. Detection performance remains stable over reduced latent resolutions, allowing the resolution to be adjusted to reduce memory consumption with minimal performance drop.

1 INTRODUCTION

Diffusion models generate samples by learning to reverse a gradual noising process, turning an initial Gaussian noise into a data-like sample through iterative denoising (Ho et al., 2020; Song et al., 2021). Text-to-image (T2I) diffusion models (Rombach et al., 2022; Ramesh et al., 2022; Ho & Salimans, 2022) extend this framework by conditioning the denoising dynamics on a text representation, commonly referred to as a text embedding, enabling high-fidelity image synthesis from natural-language prompts. Despite their success, diffusion models can exhibit memorization, where for certain prompts the models reproduce training images rather than generating genuinely novel content (Carlini et al., 2023; Somepalli et al., 2023a; Webster, 2023). Such prompt-specific reproduction raises practical concerns, including privacy leakage and copyright risks (Carlini et al., 2023; Somepalli et al., 2023b).

A representative detection method is the metric of Wen et al. (2024), which measures the difference between conditional and unconditional score. However, the measured values fluctuate substantially across initial noises and its detection accuracy remains limited, so reliable performance typically requires multi-step aggregation, where measurements are averaged across multiple denoising steps and initial noise samples to mitigate performance variability induced by the choice of initial noise. Subsequent works (Jeon et al., 2025; Asthana & Belagiannis, 2026) partially alleviate these issues, but both limitations still persist.

In this work, we focus on the score difference as the norm of a line integral over text embedding space along the linear interpolation from the unconditional embedding to the prompt embedding. Through this view, we identify that for memorized prompts, the rate of change of the conditional score along the score trajectory remains within the non-memorized range over most of the interpolation segment and rises sharply within a narrow interval near the prompt embedding, forming an abrupt transition that is associated with the anomalously large score difference observed in memorized prompts. A corresponding transition is observed in the generated image across the same interval. Below the interval, the generated image follows the same behavior as on a non-memorized interpolation and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

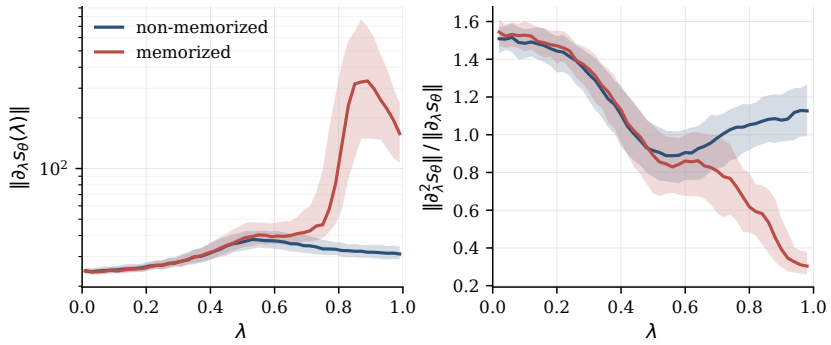


Figure 1: Score variation along the interpolation c_λ at $t = T$. First-order $\log \|\partial s_\theta(\lambda)\|$ and normalized second-order $\|\partial^2 s_\theta\|/\|\partial s_\theta\|$ of s_θ along λ at $t = T$ approximated by finite differences and measured over 100 memorized and 100 non-memorized prompts. Solid lines denote the prompt mean and shaded regions denote the prompt variance. The two prompt types behave alike at small λ and diverge sharply beyond a particular λ.

remains aligned with the prompt semantics. Across the interval, the generated image switches to the memorized training image and reproduces it across the remainder of the score trajectory.

Motivated by this localized increase, we use the Frobenius norm of the condition Jacobian evaluated at the prompt embedding at the first denoising step as a signal for memorization detection. This signal achieves state-of-the-art detection accuracy and remains stable across initial noises. This stability also extends to a range of reduced latent resolutions. Since lowering the latent resolution reduces memory consumption, selecting an appropriate resolution allows substantial memory savings with minimal loss in detection performance. Our method therefore offers a favorable trade-off between memory consumption and detection performance. Our contributions are summarized as follows.

- We express the score difference as the norm of a line integral over text embedding space and show that, for memorized prompts, the rate of change of the conditional score remains within the non-memorized range over most of the interpolation path and rises sharply within a localized region near the prompt embedding.
- We capture this localized variation through the condition Jacobian evaluated at the prompt embedding, which achieves state-of-the-art detection accuracy and remains stable across initial noises. Its robustness to reduced latent resolutions provides a favorable trade-off between memory consumption and detection performance.
- We track the generated image along the interpolation and find that both prompt types produce prompt-aligned images across the smooth segment, while memorized prompts switch to the memorized training image across the narrow interval. This narrow interval sits farther from the prompt embedding for local memorization than for global memorization, providing an embedding-space view consistent with the mitigation difficulty reported in prior work (Wen et al., 2024; Chen et al., 2025).

2 Related Work

Memorization Detection. Memorization in diffusion models is determined at the early stage of denoising. Jain et al. (2025) show that unconditional generation at the early denoising steps prevents memorization even when memorized prompt is applied afterward, so the trajectory toward the memorized image is locked in within the first few steps. At the early stage, the score difference coincides with the CFG term up to a constant, and Jin et al. (2026) show that a large CFG term at this stage sharply concentrates generations onto a narrow set of outputs. The score difference at the early stage therefore directly governs whether the trajectory converges to a memorized image, and reliable detection reduces to measuring this quantity.

Existing memorization detection methods are divided into two categories based on whether they use the score difference as the detection signal. The first category uses the score difference directly. Wen et al. (2024) first propose the score difference as a memorization detection signal, summing it

across multiple denoising steps. Jeon et al. (2025) combine the score difference with the sharpness of the predicted log-probability with respect to the initial noise. Asthana & Belagiannis (2026) extend the score difference by additionally measuring the angular alignment between the conditional and unconditional scores at the last denoising stage.

The second category does not rely on the score difference and instead detects memorization through alternative signals. Carlini et al. (2023) generate multiple images for a given prompt and examine whether highly similar outputs are produced across different noise samples. Webster (2023) performs one-step generation across multiple seeds and detects memorization through edge consistency among the outputs. Hintersdorf et al. (2024) compute pairwise SSIM between outputs generated from different seeds. Ren et al. (2024) measure the entropy of cross-attention.

Table 1: Score difference-based methods.

Method	Aux. signal	Probe
Wen (2024)	multi-step	$\{c_\phi, c\}$
Asthana (2026)	angular align.	$\{c_\phi, c\}$
Jeon (2025)	sharpness in x_T	$\{c_\phi, c\}$
Ours	J_c	near c

Text Embedding Interpolation. The behavior of text-to-image diffusion models on intermediate embeddings between two text embeddings has been studied as a tool for controllable generation. Deckers et al. (2024) show that linear interpolation between two prompt embeddings yields images whose visual attributes blend gradually along the interpolation path, He et al. (2024) extend this property to attention-level interpolation for smoother transitions across the path, and Karris et al. (2025) treat each text embedding as a point on a Wasserstein space and interpret the interpolation between two text embeddings as an optimal transport interpolation on this space.

3 Memorization Detection in Text Embedding Space

3.1 Wen Metric as a Line Integral

The Wen metric (Wen et al., 2024) quantifies memorization through the score difference

$$d_t = \|s_\theta(x_t, t, c) - s_\theta(x_t, t, c_\phi)\|_2, \tag{1}$$

where $s_\theta(x_t, t, c)$ denotes the score predicted at noise level t from the noisy latent x_t under the prompt embedding c and c_ϕ denotes the unconditional embedding. We express d_t as the norm of a line integral over text embedding space. Along the linear interpolation $c_\lambda = (1-\lambda)c_\phi + \lambda c$ for $\lambda \in [0, 1]$ the score difference is

$$s_\theta(x_t, t, c) - s_\theta(x_t, t, c_\phi) = \int_0^1 \partial_\lambda s_\theta(x_t, t, c_\lambda)\, d\lambda. \tag{2}$$

Define $r_\lambda := \|\partial_\lambda s_\theta(x_t, t, c_\lambda)\|_2$. By the triangle inequality, $d_t \leq \int_0^1 r_\lambda d\lambda$, so a large d_t requires r_λ to be large along at least part of the interpolation path. To investigate how this requirement manifests in memorized prompts, we compare r_λ along the path for memorized and non-memorized prompts, examining whether and where it becomes elevated relative to the non-memorized case when d_t is anomalously large.

Non-memorized prompts. Text-to-image diffusion models generate coherent images not only for prompts outside the training set but also for embeddings obtained by linearly interpolating between two prompt embeddings or between an unconditional embedding and a prompt embedding, with visual attributes blending gradually along the interpolation segment (He et al., 2024; Deckers et al., 2024). Figure 1 shows that, along the interpolation from the unconditional embedding c_ϕ to a non-memorized prompt embedding c, the rate of change r_λ remains relatively stable across $\lambda \in [0, 1]$, with no pronounced localized increase. The conditional score thus changes gradually along the path, rather than exhibiting a sharp transition within a particular interval.

Memorized prompts. For memorized prompts with an anomalously large d_t, the bound above requires r_λ to be large along at least part of the interpolation path. If r_λ remained approximately constant along the path, as observed for non-memorized prompts, its level would have to be high throughout the segment to accommodate such a large d_t. Figure 1, however, reveals a different

pattern. Over much of the segment, r_λ remains within the typical non-memorized range. It rises sharply only near the memorized endpoint, typically around $\lambda \approx 0.7$ in our experiments. The distinguishing feature is therefore a localized increase in r_λ near c_{mem}, rather than an elevated rate of change throughout the interpolation path. Motivated by this observation, we use the local sensitivity of the conditional score near the prompt embedding as a signal for memorization detection.

3.2 Detection via the Condition Jacobian

By the chain rule, the derivative of the conditional score along the interpolation path satisfies

$$\partial_\lambda s_\theta(x_t, t, c_\lambda) = J_c(x_t, t, c_\lambda)\,\Delta, \qquad \Delta \coloneqq c - c_\phi. \tag{3}$$

The rate of change r_λ analyzed in Section 3.1 is therefore

$$r_\lambda = \|J_c(x_t, t, c_\lambda)\,\Delta\|_2, \tag{4}$$

where

$$J_c(x_t, t, c) \coloneqq \frac{\partial s_\theta(x_t, t, c)}{\partial c} \tag{5}$$

is the condition Jacobian. The localized increase in r_λ identified in Section 3.1 corresponds to a sharp rise in $\|J_c(x_t, t, c_\lambda)\,\Delta\|_2$ near c_{mem}, motivating the use of condition sensitivity in this region as a signal for memorization detection.

A direct measurement of the underlying response is the Jacobian itself, evaluated at the early stage. The directional response $\|J_c\,\Delta\|_2$ admits the upper bound

$$\|J_c\,\Delta\|_2 \;\le\; \|J_c\|_F\,\|\Delta\|_2, \tag{6}$$

so we summarize J_c by its Frobenius norm

$$n_c(c) \;\coloneqq\; \|J_c(x_T, T, c)\|_F. \tag{7}$$

We adopt n_c as the detection signal because $\|J_c\|_F^2$ admits an unbiased Hutchinson estimator built from VJPs, which is compatible with the standard backward of the diffusion UNet, allowing the detection signal to be computed efficiently in both compute and memory. The same VJP-based estimator further extends to an input-noise sensitivity term that complements n_c, which we introduce below. This approximation is justified by the structure of J_c at memorized embeddings, where the response tends to be large not only along Δ but across a broad range of directions in text embedding space, so the Frobenius norm captures the relevant signal that the directional quantity $\|J_c\,\Delta\|_2$ measures, as we verify in Appendix F.

Complementary axis from input-noise sensitivity. Memorization also leaves a signal in the geometry of the input-noise space. Jeon et al. (2025) report that memorized prompts exhibit sharper conditional log densities and larger curvature differences between conditional and unconditional distributions, with detectable differences already present at the initial sampling stage. To express this conditioning effect, we write

$$p_T(x \mid c) = p_T(x)\,q_c(x), \qquad q_c(x) \coloneqq \frac{p_T(x \mid c)}{p_T(x)},$$

where $p_T(x)$ is the unconditional input-noise density and $q_c(x)$ is a density ratio. Here, T denotes the first finite-noise evaluation step, at which the unconditional distribution is approximately standard Gaussian. Taking the Hessian of the log density gives

$$\begin{aligned} \nabla_x^2 \log p_T(x \mid c) &= \nabla_x^2 \log p_T(x) + \nabla_x^2 \log q_c(x) \\ &\approx -I + \nabla_x^2 \log q_c(x). \end{aligned} \tag{8}$$

Thus, the additional sharpening relative to the unconditional density is encoded in the curvature of $\log q_c$. Although q_c also captures ordinary conditioning effects, this curvature provides a potential complementary signal for memorization.

Using the normalized score, we define the input-noise Jacobian and the corrected feature as

$$\begin{aligned} J_x(x_T, T, c) &\coloneqq \frac{\partial s_\theta(x_T, T, c)}{\partial x_T}, \\ n_x(c) &\coloneqq \|J_x(x_T, T, c) + I\|_F \approx \left\|\nabla_x^2 \log q_c(x_T)\right\|_F. \end{aligned} \tag{9}$$

Adding I removes the Gaussian reference curvature $-I$ to better capture the memorization signal. Consistent with this interpretation, Jeon et al. (2025) report improved early-stage memorization detection by weighting the score difference with the conditional-unconditional Hessian difference, which approximates $J_x + I$ under the Gaussian reference approximation.

The corrected Jacobian can also be obtained by differentiating the velocity prediction, which is used in SD v2.0. Given

$$x_t = a_t x_0 + \sigma_t \epsilon, \qquad v = a_t \epsilon - \sigma_t x_0, \qquad a_t^2 + \sigma_t^2 = 1, \tag{10}$$

the input noise Jacobian of the predicted velocity at the initial step is proportional to $J_x + I$.

Finally, the complementary feature can be estimated through the same VJP-based computation used for the condition feature. For a random probe with identity second moment,

$$\mathbb{E}[uu^\top] = I, \qquad n_x(c)^2 = \mathbb{E}_u \left[\left\| J_x^\top u + u \right\|_2^2 \right]. \tag{11}$$

A shared backward pass yields the VJPs for both the input noise and the condition embedding. The identity correction requires only a vector addition,

$$(J_x + I)^\top u = J_x^\top u + u. \tag{12}$$

Thus, the complementary feature requires no additional forward or backward pass beyond those used to estimate the condition feature.

For **joint feature**, w use both quantities together,

$$\phi(c) \coloneqq \big(n_x(c),\, n_c(c)\big), \tag{13}$$

and classify by a linear decision rule

$$g(n_x, n_c) = n_c - (a\, n_x + b), \tag{14}$$

where (a, b) specify a linear boundary in the (n_x, n_c) plane and a prompt is identified as memorized when $g(\phi(c)) > 0$.

4 Generated Image Through the Sharp Transition

Section 3.1 identified that r_λ rises sharply within a narrow interval along $c_\lambda = (1-\lambda)c_\phi + \lambda c$ near c. In our experiments, the sharp change in s_θ across this interval is accompanied by a corresponding sharp change in the generated image $x_0(c_\lambda)$, revealing a transition in image space as well. In image space memorization additionally splits by the spatial extent of the reproduction (Chen et al., 2025). *Global memorization* reproduces the training image over the entire generated image. *Local memorization* reproduces the training image on a region of the generated image while the remaining region varies with x_T and aligns with the prompt semantics.

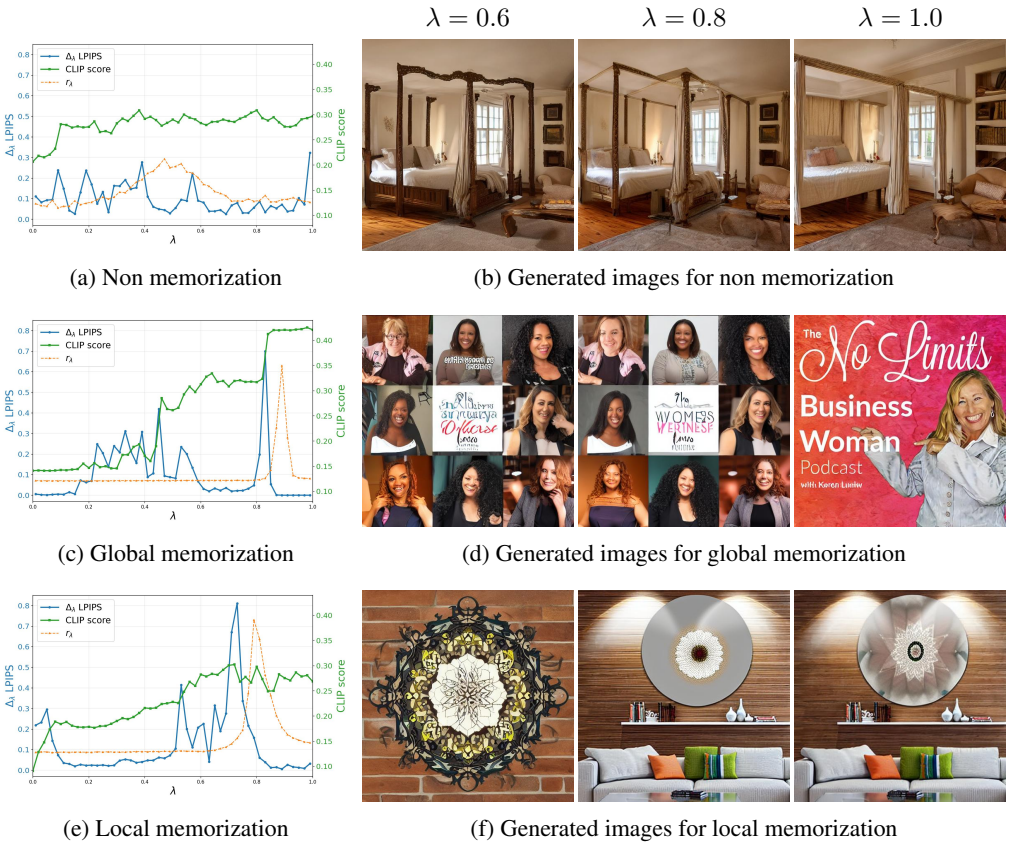
To characterize how $x_0(c_\lambda)$ varies along c_λ for each prompt type, we track three quantities on a uniform grid $\{\lambda_i\}_{i=0}^N \subset [0,1]$, where $x_0(c_\lambda)$ denotes the image obtained by running the standard sampler from x_T with c_λ held fixed across all denoising steps. We consider three quantities:

- **the terminal value of** r_λ, evaluated at $t = T$ as defined in Section 3.1
- **perceptual change**, measured by $L_i \coloneqq \mathrm{LPIPS}(x_0(c_{\lambda_{i+1}}), x_0(c_{\lambda_i}))$
- **text–image alignment**, measured by $C_i \coloneqq \mathrm{CLIP}(x_0(c_{\lambda_i}), c)$.

Behavior across the transition. The sharp rise in r_λ and the switch of $x_0(c_\lambda)$ to the training image occur at the same λ, so the location of the transition is summarized by

$$\lambda^* \coloneqq \arg\max_{\lambda \in [0,1]} r_\lambda. \tag{15}$$

Up to λ^*, the three prompt types in Figure 2 share the same trajectory. Near c_ϕ, C_i rises rapidly and L_i remains large as the generated image moves from arbitrary images toward prompt-aligned images. Over the subsequent segment, C_i and L_i stabilize and the generated image varies marginally with λ.



(a) Non memorization (b) Generated images for non memorization

(c) Global memorization (d) Generated images for global memorization

(e) Local memorization (f) Generated images for local memorization

Figure 2: (a, c, e) r_λ, inter-image LPIPS L_i, and prompt-image CLIP score C_i along $\lambda \in [0, 1]$ for a non-memorized prompt (a), a globally memorized prompt (c), and a locally memorized prompt (e). (b, d, f) Generated images $x_0(c_\lambda)$ at $\lambda \in \{0.6, 0.8, 1.0\}$ for the same prompts. The three prompt types share the same trajectory up to λ^*, where memorized prompts depart from the non-memorized regime through a peak in r_λ, a spike in L_i, and a switch of $x_0(c_\lambda)$ to the memorized training image.

For the non-memorized prompt this segment extends to $\lambda = 1$ and the generated image in Figure 2b stays visually consistent across the λ.

For globally and locally memorized prompts, the trajectory departs from this regime at λ^*. r_λ peaks, L_i exhibits a localized spike, and the generated image switches to the memorized image, which is then reproduced for all larger λ. The two memorization types diverge in post-transition behavior. Under global memorization, the memorized image occupies the entire image region, L_i returns to small values, and the generated image remains visually unchanged beyond λ^*, as shown in Figure 2d. Under local memorization, the memorized region stays fixed while the remaining region continues to vary with λ, so L_i stays non-negligible and the generated image keeps changing outside the memorized region, as shown in Figure 2f.

Location of the transition. Figure 3 reports the distribution of λ^* over locally and globally memorized prompts. Global memorization places λ^* near 1, while local memorization peaks at $\lambda^* \approx 0.8$, so the transition for local memorization sits farther from c. Since $x_0(c_\lambda)$ reproduces the training image for $\lambda > \lambda^*$, the subset of c_λ that reproduces the training image is $[\lambda^*, 1]$ and is wider for local memorization than for global memorization. This wider region in text embed-

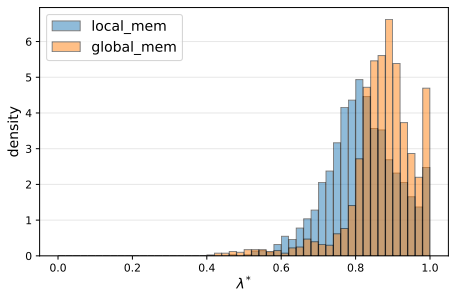


Figure 3: Distribution of λ^* on SD v1.4 over locally and globally memorized prompts. Local memorization yields smaller λ^*.

Table 2: Comparison of memorization detection methods on SD v1.4 and SD v2.0. We generate 100 samples using 100 different seeds. For $n = 1$, each sample is used independently to compute a detection score. For $n = 4$, every four samples are grouped and their scores are averaged. Mean $\pm$ standard deviation are reported over the resulting scores. Best results are shown in **bold**.

Method	SD v1.4		SD v2.0	
	AUC ↑	TPR@1%FPR ↑	AUC ↑	TPR@1%FPR ↑
$n = 1$				
Ren et al. (2024)	0.899 ± 0.018	0.263 ± 0.056	0.869 ± 0.011	0.000 ± 0.001
Wen et al. (2024)	0.947 ± 0.032	0.746 ± 0.133	0.941 ± 0.034	0.576 ± 0.270
Hintersdorf et al. (2024)	0.782 ± 0.013	0.363 ± 0.025	0.915 ± 0.011	0.488 ± 0.069
Jeon et al. (2025)	0.966 ± 0.032	0.824 ± 0.111	0.923 ± 0.078	0.552 ± 0.273
Asthana & Belagiannis (2026)	0.947 ± 0.032	0.748 ± 0.134	0.946 ± 0.033	0.600 ± 0.271
Ours(J_c only)	0.994 ± 0.003	0.959 ± 0.028	0.989 ± 0.006	0.874 ± 0.137
Ours(J_c with J_x)	$\mathbf{0.998 \pm 0.001}$	$\mathbf{0.980 \pm 0.012}$	$\mathbf{0.997 \pm 0.003}$	$\mathbf{0.971 \pm 0.028}$
$n = 4$				
Ren et al. (2024)	0.903 ± 0.007	0.248 ± 0.030	0.870 ± 0.006	0.000 ± 0.000
Wen et al. (2024)	0.986 ± 0.006	0.889 ± 0.055	0.976 ± 0.008	0.828 ± 0.070
Hintersdorf et al. (2024)	0.934 ± 0.004	0.523 ± 0.012	0.985 ± 0.001	0.873 ± 0.007
Jeon et al. (2025)	0.994 ± 0.005	0.965 ± 0.019	0.975 ± 0.013	0.812 ± 0.044
Asthana & Belagiannis (2026)	0.985 ± 0.008	0.896 ± 0.039	0.977 ± 0.006	0.848 ± 0.055
Ours(J_c only)	0.999 ± 0.001	0.992 ± 0.005	0.996 ± 0.002	0.976 ± 0.008
Ours(J_c with J_x)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.995 \pm 0.002}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.995 \pm 0.004}$

ding space aligns with prior reports that mitigation methods which suppress localized memorization signals in cross-attention (Hintersdorf et al., 2024) and methods which steer x_T away from initial noises that trigger memorization (Jeon et al., 2025; Asthana & Belagiannis, 2026) both perform worse on local memorization (Chen et al., 2025), providing an embedding-space view that is consistent with this gap.

5 Experiments

In this section, we present our experimental results organized into two main phases. First, we evaluate memorization detection performance on SD v1.4 and v2.0 (Rombach et al., 2022) to align with established evaluation protocols (Sec. 5.1). To ensure a more reliable evaluation, we additionally verify the detection performance on unseen prompts using a pre-determined decision boundary. Second, we analyze the computational efficiency of our proposed approach specifically within the context of the memorization detection task (Sec. 5.2).

5.1 Memorization Detection

Experiment Setup. We adopt the same evaluation setting as recent memorization detection studies (Jeon et al., 2025; Asthana & Belagiannis, 2026; Wen et al., 2024). For memorized prompts, we use the sets curated by Webster (2023), which contain 500 and 219 prompts for SD v1.4 and v2.0 (Rombach et al., 2022), respectively. The non-memorized counterpart consists of 500 prompts drawn from four diverse sources, namely COCO (Lin et al., 2014), Lexica (Shen et al., 2024), Tuxemon (HuggingFace, 2024), and GPT-4 (OpenAI et al., 2024). All methods are evaluated under two detection regimes, single-sample ($n = 1$) and four-sample averaging ($n = 4$), and compared using the Area Under the ROC Curve (AUC) and True Positive Rate at 1% False Positive Rate (TPR@1%FPR) on a single NVIDIA A40 GPU. Baselines include the cross-attention approach of Ren et al. (2024), the prediction discrepancy method of Wen et al. (2024), the sharpness-based method of Jeon et al. (2025), the anisotropy-based method of Asthana & Belagiannis (2026), and the pairwise SSIM method of Hintersdorf et al. (2024). Unlike existing approaches that rely on a

single scalar score, our method produces a two-dimensional feature, which precludes direct AUC computation. To enable a fair comparison, we fit a logistic regression decision boundary and project the joint feature onto the direction orthogonal to this boundary, yielding a single scalar score. All results in this subsection are reported using these projected scores.

Results. Table 2 reports detection performance on SD v1.4 and v2.0 under single sample ($n = 1$) and four sample averaging ($n = 4$). The J_c only variant already exceeds all baselines on both metrics across both models. On SD v1.4 at $n = 1$ it attains 0.994 and 0.959 against the best baseline values 0.966 and 0.824 of Jeon et al. (2025). On SD v2.0 at $n = 1$ it attains 0.989 and 0.874 against the best baseline values 0.946 and 0.600 of Asthana & Belagiannis (2026). The margin is wider in the high confidence regime, which reflects that score difference based baselines saturate below 1 at low FPR while J_c separates the two classes in this regime.

Adding J_x raises both metrics to near saturation. On SD v1.4 the joint feature reaches 0.998 and 0.980 at $n = 1$ and 1.000 and 0.995 at $n = 4$. On SD v2.0 it reaches 0.997 and 0.971 at $n = 1$ and 1.000 and 0.995 at $n = 4$. The gain over the J_c only variant is largest in the high confidence regime on SD v2.0, where the score rises from 0.874 to 0.971 at $n = 1$, indicating that J_x supplies a complementary axis when J_c alone leaves residual overlap in this regime. A residual gap between SD v1.4 and SD v2.0 remains in our method, with SD v2.0 trailing SD v1.4 in the high confidence regime at $n = 1$.

We also conduct memorization detection experiments on FLUX.1, extending our evaluation to a transformer-based architecture (Appendix A). Since no established set of memorized prompts is available for this model, we deliberately induce memorization through LoRA fine-tuning before evaluating detection performance.

Threshold Transferability. Beyond score-based metrics such as AUC, it is also important to test whether a decision threshold calibrated on one subset of prompts generalizes to held-out prompts, as evaluated by binary classification metrics such as accuracy and F-score. However, prior work has primarily reported score-based metrics such as AUC and has not explicitly evaluated this threshold transferability. To address this, we conduct a threshold transferability experiment on both SD v1.4 and SD v2.0 using the prompt sets described above. We calibrate the classification threshold on a randomly selected 20% subset of prompts and evaluate on the remaining 80%, repeating this procedure 10 times with different splits and reporting mean $\pm$ standard deviation.

Across both models and both aggregation settings, our method attains the best accuracy and F-score among all baselines, indicating that the decision boundary generalizes reliably to held-out prompts. As a representative example, on SD v1.4 with $n = 1$, our method achieves an F-score of 0.979 ± 0.003, while the strongest baseline reaches 0.898 ± 0.003. The same conclusion holds for SD v1.4 with $n = 4$ and for SD v2.0 with both $n = 1$ and $n = 4$, where our method remains the top performer across metrics. Full results for all methods, models, and settings are provided in Table 6 and Table 7 in Appendix C.

5.2 Favorable Memory Performance Trade-off

Performance across Latent Resolutions. In UNet-based latent diffusion models such as Stable Diffusion, the fully convolutional architecture allows the latent resolution to be adjusted at inference time. Reducing the latent resolution lowers memory consumption and accelerates inference, but typically degrades generation quality and, for existing detection methods, reduces detection performance.

Table 3 reports the AUC and TPR@1%FPR of each method as a function of latent resolution on SD v1.4 and SD v2.0, with scores averaged over $n = 4$ samples. The effect of reducing latent resolution varies across methods and models.

On SD v2.0, the joint feature based on J_c and $J_x + I$ maintains strong detection performance across all tested resolutions, achieving an AUC of at least 0.987 and a TPR@1%FPR of at least 0.908. Our method outperforms all baselines on both metrics at every tested resolution. Even at the smallest resolution, $4 \times 8 \times 8$, it achieves an AUC of 0.987 and a TPR@1%FPR of 0.908, exceeding the best baseline results at the default resolution, $4 \times 96 \times 96$ (0.985 and 0.873, respectively).

Table 3: Comparison of memorization detection methods across reduced latent resolutions on SD v1.4 and SD v2.0. r denotes the spatial side of the latent so that the latent resolution is $4 \times r \times r$ with default $r = 64$ for SD v1.4 and $r = 96$ for SD v2.0. We generate 100 samples using 100 different seeds. Every four samples are grouped and their scores are averaged and the mean over the resulting scores is reported. Best results per row are shown in bold.

SD v1.4						
AUC	Ren (2024)	Wen (2024)	Hintersdorf (2024)	Jeon (2025)	Asthana (2026)	**Ours**
$r = 8$	0.567	**0.884**	0.559	0.821	0.828	0.805
$r = 16$	0.719	0.951	0.401	0.935	0.893	**0.956**
$r = 32$	0.837	0.779	0.514	0.691	0.627	**0.996**
$r = 64$	0.903	0.986	0.934	0.994	0.985	**1.000**
TPR@1%FPR						
$r = 8$	0.002	**0.489**	0.125	0.349	0.167	0.328
$r = 16$	0.011	0.652	0.068	0.546	0.441	**0.741**
$r = 32$	0.054	0.238	0.155	0.045	0.106	**0.982**
$r = 64$	0.248	0.889	0.523	0.965	0.896	**0.995**

SD v2.0						
AUC	Ren (2024)	Wen (2024)	Hintersdorf (2024)	Jeon (2025)	Asthana (2026)	**Ours**
$r = 8$	0.784	0.967	0.507	0.946	0.966	**0.987**
$r = 16$	0.747	0.982	0.560	0.942	0.982	**0.996**
$r = 32$	0.935	0.989	0.453	0.920	0.989	**0.994**
$r = 64$	0.932	0.923	0.794	0.830	0.922	**0.999**
$r = 96$	0.870	0.976	0.985	0.975	0.977	**1.000**
TPR@1%FPR						
$r = 8$	0.025	0.621	0.015	0.569	0.616	**0.908**
$r = 16$	0.001	0.881	0.022	0.697	0.880	**0.969**
$r = 32$	0.438	0.929	0.010	0.269	0.927	**0.956**
$r = 64$	0.634	0.336	0.243	0.081	0.333	**0.989**
$r = 96$	0.000	0.828	0.873	0.812	0.848	**0.995**

On SD v1.4, our method retains near-default performance at $4 \times 32 \times 32$, achieving an AUC of 0.996 and a TPR@1%FPR of 0.982, compared with 1.000 and 0.995 at the default resolution. Performance decreases at smaller resolutions, although our method still leads all baselines on both metrics at $4 \times 16 \times 16$. At $4 \times 8 \times 8$, its AUC and TPR@1%FPR fall to 0.805 and 0.328, and Wen et al.'s method performs best. These results indicate that the extent of resolution reduction supported by our detector depends on the model. These results indicate that the extent of resolution reduction supported by our detector depends on the model.

Reducing the latent resolution lowers the memory required for detection. By selecting an appropriate latent resolution for each model, our method reduces memory consumption while minimizing performance loss, offering a favorable trade-off between memory consumption and detection performance.

6 LIMITATION

Our method achieves state-of-the-art memorization detection accuracy and offers a favorable trade-off between memory consumption and detection performance through an appropriate choice of latent resolution. Two aspects remain open. We identify the sharp rise of the conditional score along the interpolation near the prompt embedding as the source of the anomalously large score difference on memorized prompts but we do not characterize why the conditional score departs from the smooth variation observed on non-memorized prompts within this interval. We track the generated image along the interpolation and locate the switch to the memorized training image but we do not translate this behavior of the generated image into a mitigation procedure that suppresses the reproduction of the memorized training image at the prompt embedding.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on the design of our experiments and to implement parts of the experimental code. We have not used generative AI tools to propose or refine hypotheses, to generate synthetic datasets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims or assist in proving them, or to interpret results, and the remaining required disclosure tasks (translation, dataset cleaning and reformatting, qualitative data analysis) are not applicable to this work. Additionally, we used generative AI tools to edit the manuscript for grammar and readability. We have reviewed all AI-assisted work. Experimental designs suggested by AI tools were evaluated and finalized by the authors, all AI-generated code was reviewed line by line by one of the authors and tested for correctness before use, and all edited text was compared with the original draft to confirm that no technical content was altered. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies memorization in text-to-image diffusion models and develops a detector to support auditing for potential privacy leakage and unintended reproduction of copyrighted content. Our experiments use existing pretrained models and prompt sets described in Section 5 and Appendix D. Improved detection may help researchers and model developers identify prompts associated with training-image reproduction and prioritize further investigation and mitigation. The method also presents a dual-use risk: identifying such prompts could facilitate attempts to extract memorized training images. We therefore encourage its use for authorized auditing and responsible research, with appropriate care when handling or sharing reproduced content. Detection results are subject to false positives and false negatives, and their reliability depends on the model, prompt distribution, and calibration setting. They should be interpreted as empirical indicators of memorization risk. Assessing specific cases requires additional evidence, and protecting against training-data disclosure requires complementary mitigation measures.

ACKNOWLEDGMENTS

Use unnumbered first level headings for the acknowledgments. All acknowledgments go at the end of the paper before the list of references. Moreover, you are required to declare funding (financial activities supporting the submitted work) and competing interests (related financial activities outside the submitted work). More information about this disclosure can be found at: https://neurips.cc/Conferences/2026/PaperInformation/FundingDisclosure.

Do **not** include this section in the anonymized submission, only in the final paper. You can use the ack environment provided in the style file to automatically hide this section in the anonymized submission.

REFERENCES

Rohan Asthana and Vasileios Belagiannis. Detecting and mitigating memorization in diffusion models through anisotropy of the log-probability, 2026. URL https://arxiv.org/abs/2601.20642.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In Joseph A. Calandrino and Carmela Troncoso (eds.), *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pp. 5253–5270. USENIX Association, 2023. URL https://www.usenix.org/conference/usenixsecurity23/presentation/carlini.

Chen Chen, Daochang Liu, Mubarak Shah, and Chang Xu. Exploring local memorization in diffusion models via bright ending attention. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 29827–29841, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/498c7e2589cd2e5b9886d0ba9a5e6032-Paper-Conference.pdf.

Niklas Deckers, Julia Peters, and Martin Potthast. Manipulating embeddings of stable diffusion prompts. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/845. URL https://doi.org/10.24963/ijcai.2024/845.

Qiyuan He, Jinghao Wang, Ziwei Liu, and Angela Yao. Aid: Attention interpolation of text-to-image diffusion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 97766–97799. Curran Associates, Inc., 2024. doi: 10.52202/079017-3101. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b12a1d1014e952e676f5d6931d03241a-Paper-Conference.pdf.

Dominik Hintersdorf, Lukas Struppek, Kristian Kersting, Adam Dziedzic, and Franziska Boenisch. Finding nemo: Localizing neurons responsible for memorization in diffusion models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 88236–88278. Curran Associates, Inc., 2024. doi: 10.52202/079017-2800. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a102dd5931da01e1b40205490513304c-Paper-Conference.pdf.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

HuggingFace. Tuxemon. https://huggingface.co/datasets/diffusers/tuxemon, 2024. Accessed: 2026-02-03.

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. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12871–12879, Los Alamitos, CA, USA, June 2025. IEEE Computer Society. doi: 10.1109/CVPR52734.2025.01201. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52734.2025.01201.

Dongjae Jeon, Dueun Kim, and Albert No. Understanding and mitigating memorization in generative models via sharpness of probability landscapes. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27091–27112. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/jeon25a.html.

Cheng Jin, Qitan Shi, and Yuantao Gu. Stage-wise dynamics of classifier-free guidance in diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fP0s1TEow3.

Nicholas Karris, Luke Durell, Javier Flores, and Tegan Emerson. Which way from b to a: The role of embedding geometry in image interpolation for stable diffusion. *arXiv preprint arXiv:2511.12757*, 2025.

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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

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 Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 340–356, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-72980-5.

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 (CVPR)*, pp. 10684–10695, June 2022.

Xinyue Shen, Yiting Qu, Michael Backes, and Yang Zhang. Prompt Stealing Attacks Against Text-to-Image Generation Models. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2024.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47783–47803. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9521b6e7f33e039e7d92e23f5e37bbf4-Paper-Conference.pdf.

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 (CVPR)*, pp. 6048–6058, June 2023b.

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.

Ryan Webster. A reproducible extraction of training images from diffusion models, 2023. URL https://arxiv.org/abs/2305.08694.

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.

Table 4: Memorization detection performance on the LoRA-fine-tuned FLUX.1-dev model. Results are based on four random seeds. Latent dimensions are reported as channels × height × width. Best results are shown in bold.

Method	AUC ↑	TPR@1%FPR ↑
Ours ($16 \times 64 \times 64$)	**0.997**	**0.892**
Ours ($16 \times 32 \times 32$)	0.981	0.734
Ours ($16 \times 16 \times 16$)	0.961	0.519
Wen et al. (2024)	0.991	0.443
Asthana & Belagiannis (2026)	0.991	0.443
Hintersdorf et al. (2024)	0.628	0.228

A Memorization Detection on a Transformer-Based Model

We extend the experiments in Sec. 5 to a transformer-based text-to-image model to examine whether our detection method remains effective beyond UNet-based architectures. Specifically, we evaluate a fine-tuned FLUX model at the default and reduced latent resolutions, assessing both detection performance and its sensitivity to spatial resolution.

Experimental setup. In the absence of an available public memorized-prompt benchmark for this model family in our experimental setting, we construct a controlled memorization setting by fine-tuning `black-forest-labs/FLUX.1-dev` with low-rank adaptation (LoRA). We use 158 prompt–image pairs and train for 40,000 steps to induce memorization. We then evaluate detection performance using AUC and TPR@1%FPR, as in Sec. 5.1. The results reported here are based on four random seeds and should therefore be regarded as a preliminary evaluation compared with the 100-seed experiments on SD v1.4 and SD v2.0.

Reduced latent resolutions and positional coordinates. We evaluate our method with latent tensors of size $16 \times 64 \times 64$, $16 \times 32 \times 32$, and $16 \times 16 \times 16$, where the first dimension denotes the number of latent channels. The $16 \times 64 \times 64$ setting serves as the default resolution in this experiment.

When reducing the latent resolution, we also adjust the spatial coordinates used for rotary positional embeddings. Let $H_0 \times W_0$ denote the default image-token grid and let q be the reduction factor along each spatial dimension. On the reduced grid of size $(H_0/q) \times (W_0/q)$, the spatial coordinates are assigned with stride q:

$$(i, j) \longmapsto (qi, qj), \qquad 0 \le i < H_0/q, \quad 0 \le j < W_0/q. \tag{16}$$

Thus, the default, half-resolution, and quarter-resolution settings use spatial coordinate strides of 1, 2, and 4, respectively. This assignment represents the reduced grid as a coarser sampling of the default positional coordinate system.

In FLUX, image and text tokens participate jointly in attention. Reducing each spatial dimension by a factor of q reduces the number of image tokens by a factor of q^2, shortening the joint attention sequence and reducing its attention-memory requirements. This provides a mechanism for reducing detection memory in transformer-based models, complementing the latent-resolution experiments on UNet-based models in Sec. 5.2.

Baseline applicability. We compare against Wen et al. (2024), Asthana & Belagiannis (2026), and Hintersdorf et al. (2024). The cross-attention entropy detector of Ren et al. (2024) is not directly applicable in its original form because FLUX jointly attends over image and text tokens rather than using the separate cross-attention mechanism assumed by that detector. Jeon et al. (2025) could not be evaluated because its execution exceeded the available GPU memory.

Results. Table 4 shows that our method achieves the highest AUC and TPR@1%FPR at the default latent resolution. Its AUC reaches 0.997, compared with 0.991 for the strongest evaluated baselines, while its TPR@1%FPR reaches 0.892, exceeding their value of 0.443 by 44.9 percentage points.

Table 5: Effect of the number of Hutchinson iterations K on detection performance. K denotes the number of random vectors used to estimate each Frobenius norm. All results use the joint feature space. Mean $\pm$ standard deviation are reported over the resulting scores. Best results are shown in **bold**.

K	SD v1.4		SD v2.0	
	AUC ↑	TPR@1%FPR ↑	AUC ↑	TPR@1%FPR ↑
		$n = 1$		
$K = 1$	0.994 ± 0.003	0.960 ± 0.028	0.995 ± 0.004	0.934 ± 0.067
$K = 2$	0.996 ± 0.002	0.972 ± 0.019	0.997 ± 0.003	0.955 ± 0.049
$K = 4$	$\mathbf{0.998 \pm 0.001}$	$\mathbf{0.980 \pm 0.012}$	$\mathbf{0.997 \pm 0.003}$	$\mathbf{0.971 \pm 0.028}$
		$n = 4$		
$K = 1$	0.999 ± 0.001	0.992 ± 0.004	1.000 ± 0.000	0.990 ± 0.007
$K = 2$	0.999 ± 0.001	0.994 ± 0.002	1.000 ± 0.000	0.994 ± 0.005
$K = 4$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.995 \pm 0.002}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.995 \pm 0.004}$

Reducing the latent resolution introduces a measurable performance trade-off. At $16 \times 32 \times 32$, our method achieves an AUC of 0.981 and a TPR@1%FPR of 0.734. At $16 \times 16 \times 16$, these values decrease to 0.961 and 0.519, respectively. Although the reduced-resolution AUC values fall below those of Wen et al. (2024) and Asthana & Belagiannis (2026), TPR@1%FPR remains higher than that of every evaluated baseline at both reduced resolutions.

These results provide evidence that our detection method extends to a transformer-based model under induced memorization. They also show that reducing the latent resolution preserves a useful detection signal, although performance at low FPR is more sensitive to resolution reduction than AUC. The appropriate resolution therefore depends on the desired balance between attention-memory requirements and detection sensitivity.

B Effect of Hutchinson Estimator Iteration

Our detector estimates $\|J_x\|_F$ and $\|J_c\|_F$ using a VJP-based Hutchinson estimator. For a matrix $J \in \mathbb{R}^{m \times n}$, the estimator approximates

$$\|J\|_F^2 = \mathbb{E}\big[\|J^\top v\|_2^2\big]$$

by averaging over K i.i.d. Gaussian random vectors v. Each iteration requires one vector-Jacobian product and is therefore much cheaper than materializing the full Jacobian. We next examine how K affects detection performance.

Table 5 shows that increasing K consistently improves both AUC and TPR@1%FPR. At $n = 1$, the gain is clear on both models. On SD v1.4, AUC increases from 0.994 to 0.998 and TPR@1%FPR from 0.960 to 0.980. On SD v2.0, the improvement is larger in the high-confidence regime, where TPR@1%FPR rises from 0.934 to 0.971. These gains are expected: larger K reduces the variance of the Hutchinson estimator, yielding more accurate Frobenius norm estimates and more stable joint features.

The effect is also reflected in the standard deviations. On SD v2.0 at $n = 1$, the standard deviation of TPR@1%FPR decreases from 0.067 at $K = 1$ to 0.028 at $K = 4$, showing that larger K makes detection outcomes more consistent across runs. This is particularly desirable in practice, where stability matters in addition to mean accuracy.

At $n = 4$, the differences become smaller because averaging over multiple noise samples already suppresses part of the estimation noise. Even so, $K = 4$ still yields measurable improvements over smaller K, especially in TPR@1%FPR. We therefore use $K = 4$ in all main experiments as a favorable balance between additional VJP cost and improved accuracy and stability.

Table 6: Threshold transferability results for Accuracy and Precision. We use 20% of prompts for threshold calibration and 80% for testing, and report mean ± standard deviation over 10 random splits.

Method	SD v1.4			
	$N = 1$		$N = 4$	
	Acc.	Prec.	Acc.	Prec.
Ren	0.835 ± 0.007	0.828 ± 0.008	0.841 ± 0.008	0.830 ± 0.011
Wen	0.881 ± 0.003	0.939 ± 0.012	0.945 ± 0.004	0.973 ± 0.010
Hintersdorf	0.707 ± 0.005	0.794 ± 0.021	0.784 ± 0.006	0.818 ± 0.016
Jeon	0.902 ± 0.002	0.954 ± 0.004	0.961 ± 0.002	0.975 ± 0.003
Asthana	0.881 ± 0.003	0.939 ± 0.012	0.946 ± 0.004	0.972 ± 0.011
Ours	$\mathbf{0.979 \pm 0.003}$	$\mathbf{0.996 \pm 0.001}$	$\mathbf{0.989 \pm 0.002}$	$\mathbf{1.000 \pm 0.000}$

Method	SD v2.0			
	$N = 1$		$N = 4$	
	Acc.	Prec.	Acc.	Prec.
Ren	0.827 ± 0.009	0.667 ± 0.028	0.827 ± 0.011	0.667 ± 0.031
Wen	0.875 ± 0.004	0.822 ± 0.026	0.938 ± 0.005	0.912 ± 0.029
Hintersdorf	0.871 ± 0.005	0.782 ± 0.015	0.938 ± 0.004	0.916 ± 0.020
Jeon	0.850 ± 0.002	0.830 ± 0.022	0.915 ± 0.003	0.911 ± 0.021
Asthana	0.852 ± 0.009	0.906 ± 0.013	0.918 ± 0.006	0.956 ± 0.021
Ours	$\mathbf{0.976 \pm 0.005}$	$\mathbf{0.977 \pm 0.008}$	$\mathbf{0.988 \pm 0.007}$	$\mathbf{0.993 \pm 0.007}$

Table 7: Threshold transferability results for Recall and F1-score. We use 20% of prompts for threshold calibration and 80% for testing, and report mean ± standard deviation over 10 random splits.

Method	SD v1.4			
	$N = 1$		$N = 4$	
	Rec.	F1	Rec.	F1
Ren	0.847 ± 0.018	0.837 ± 0.008	0.858 ± 0.026	0.843 ± 0.010
Wen	0.829 ± 0.014	0.870 ± 0.005	0.917 ± 0.014	0.943 ± 0.005
Hintersdorf	0.561 ± 0.002	0.656 ± 0.015	0.732 ± 0.030	0.772 ± 0.011
Jeon	0.858 ± 0.008	0.898 ± 0.003	0.959 ± 0.005	0.964 ± 0.002
Asthana	0.820 ± 0.015	0.870 ± 0.005	0.918 ± 0.016	0.944 ± 0.005
Ours	$\mathbf{0.962 \pm 0.006}$	$\mathbf{0.979 \pm 0.003}$	$\mathbf{0.978 \pm 0.005}$	$\mathbf{0.989 \pm 0.003}$

Method	SD v2.0			
	$N = 1$		$N = 4$	
	Rec.	F1	Rec.	F1
Ren	0.878 ± 0.051	0.756 ± 0.006	0.879 ± 0.059	0.756 ± 0.009
Wen	0.784 ± 0.037	0.787 ± 0.010	0.886 ± 0.021	0.898 ± 0.006
Hintersdorf	0.804 ± 0.013	0.792 ± 0.006	0.879 ± 0.018	0.897 ± 0.006
Jeon	0.686 ± 0.035	0.718 ± 0.015	0.852 ± 0.027	0.871 ± 0.005
Asthana	0.792 ± 0.034	0.833 ± 0.017	0.879 ± 0.024	0.914 ± 0.007
Ours	$\mathbf{0.944 \pm 0.023}$	$\mathbf{0.958 \pm 0.010}$	$\mathbf{0.967 \pm 0.026}$	$\mathbf{0.980 \pm 0.012}$

C THRESHOLD TRANSFERABILITY

As discussed in Sec. 5.1, existing memorization detection evaluations typically compute scores for all prompts, determine an optimal threshold on the entire dataset, and report performance at that threshold. While informative, this protocol does not assess whether the selected threshold generalizes to unseen prompts. In practice, a deployed detector must determine its threshold from a limited calibration set and apply it to previously unseen queries.

To evaluate this setting, we split the prompt set into 20% for calibration and 80% for testing, repeat the procedure 10 times with different random splits, and report the mean $\pm$ standard deviation of Accuracy, Precision, Recall, and F-score. For baseline methods that produce a single scalar detection score, we determine the binary decision threshold by maximizing Youden's J statistic ($J = \mathrm{TPR} - \mathrm{FPR}$) on the calibration split. For our method, which produces a two-dimensional feature, we fit a logistic regression decision boundary on the calibration split and apply it to the test split.

Tables 6 and 7 report the results on SD v1.4 and SD v2.0, respectively. Our method achieves the best performance across all four metrics in both the single-sample ($N = 1$) and four-sample ($N = 4$) settings on both models. On SD v1.4 with $N = 1$, our method attains an F-score of 0.979 ± 0.003, compared to 0.898 ± 0.003 for the strongest baseline, Jeon (2025). With $N = 4$, the difference remains substantial, with our method reaching 0.989 ± 0.003 versus 0.964 ± 0.002.

On SD v2.0, where detection is more challenging overall, the advantage of the joint feature space becomes even clearer. With $N = 1$, our method achieves an F-score of 0.958 ± 0.010, substantially outperforming the best baseline, Asthana and Belagiannis (2026), at 0.833 ± 0.017. With $N = 4$, our method further improves to 0.980 ± 0.012, while the best baseline reaches 0.914 ± 0.007. The consistently low standard deviations across random splits indicate that the decision boundary learned from the joint feature space transfers reliably to held-out prompts.

D Detection Performance Across Time Steps

This appendix examines whether the our method signal persists across the entire denoising trajectory, rather than being confined to a single time step. In the Sec. 5.1, all detection results are reported at the first denoising step ($t = T$). Here we evaluate our method independently at each time step t along the DDIM schedule, computing AUC and TPR@1%FPR with $n = 4$ samples on both SD v1.4 and SD v2.0.

Fig. 4 reports the results. On SD v1.4, both AUC and TPR@1%FPR remain consistently high across the entire denoising trajectory. AUC stays above 0.99 at all time steps, and TPR@1%FPR remains above 0.97, with narrow confidence bands throughout. This demonstrates that the discriminative power of the joint feature space is not confined to a particular noise level but persists as a stable property throughout the generation process.

On SD v2.0, AUC also exhibits strong stability, remaining above 0.93 across all time steps. TPR@1%FPR, however, shows a gradual decrease as t approaches zero, declining from approximately 0.99 at $t = T$ to around 0.74 at $t \approx 0$. While the detection signal remains meaningful at all time steps, the separation in the high confidence regime becomes less pronounced at later stages of generation.

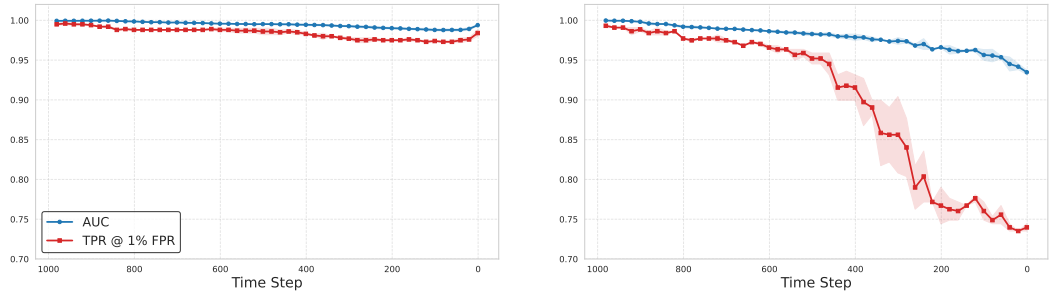
Overall, these results indicate that the heightened local sensitivity of memorized prompts is not a transient phenomenon tied to a specific noise level, but rather a persistent characteristic that the denoiser exhibits throughout the generation process.

E Detection on Another Model

To evaluate whether our joint detection method generalizes beyond the Stable Diffusion models, we conduct additional experiments on Realistic Vision v5.1[1], a community fine-tuned model based on Stable Diffusion v1.5. We use the memorized prompt set provided by Webster (2023) and follow the same evaluation protocol as in Sec. 5.1, computing detection scores under both single sample ($n = 1$) and four sample averaging ($n = 4$) settings.

Table 8 reports the results. Our method achieves an AUC of 0.964 ± 0.007 and TPR@1%FPR of 0.804 ± 0.064 with a single sample, improving to an AUC of 0.976 ± 0.002 and TPR@1%FPR of 0.884 ± 0.004 with four sample averaging. These results demonstrate that the joint signal transfers effectively to a fine-tuned model without any modification to the detection method, suggesting that the relationship between local sensitivity and memorization is not specific to the original Stable Diffusion training but persists across model variants.

[1] https://civitai.com/



(a) SD v1.4 (b) SD v2.0

Figure 4: Detection performance of our joint method across denoising time steps. Scores are computed by averaging over $n = 4$ samples, and the procedure is repeated 25 times to obtain the mean and standard deviation.

Table 8: Memorization detection performance on Realistic Vision v5.1. Mean $\pm$ standard deviation are reported over the resulting scores.

Method	$N = 1$		$N = 4$	
	AUC ↑	TPR@1%FPR ↑	AUC ↑	TPR@1%FPR ↑
Ours	0.964 ± 0.007	0.804 ± 0.064	0.976 ± 0.002	0.884 ± 0.004

F Additional Analysis on Condition Jacobian

This appendix provides supplementary evidence for the analysis in Sec. 3.2, where we showed that the memorization signal in the condition Jacobian, as sensitivity, is concentrated near the target condition c rather than distributed along the entire interpolation path from $c_\emptyset$ to c.

In Sec. 3.2, we further argued that the Frobenius norm $\|J_c\|_F$ serves as a reasonable proxy for the directional sensitivity $\|J_c \cdot \Delta\|_2$. A natural concern is whether the heightened sensitivity of memorized prompts is concentrated specifically along the $\Delta = c - c_\emptyset$ direction, in which case $\|J_c\|_F$ would be a loose upper bound and a poor proxy. To investigate this, we examine two comparisons on both SD v1.4 and SD v2.0.

First, Fig. 6a and Fig. 6c compare the directional sensitivity along $\hat{\Delta} = \Delta/\|\Delta\|_2$ with the average sensitivity along 20 random unit directions. Specifically, for each prompt we sample 20 independent random vectors $v_1, \ldots, v_{20}$ uniformly on the unit sphere in the condition embedding space, compute $\|J_c \cdot v_k\|_2$ for each, and take their average. If the sensitivity were concentrated entirely along Δ, memorized prompts would appear far above the $y = x$ line, since $\|J_c \cdot \hat{\Delta}\|_2$ would be much larger than the average over random directions. On SD v1.4, memorized prompts lie relatively close to the $y = x$ line, indicating that the sensitivity increase is distributed fairly uniformly across directions rather than concentrated along Δ. On SD v2.0, memorized prompts tend to lie further above the $y = x$ line, suggesting a relatively stronger concentration along the Δ direction compared to SD v1.4. Nevertheless, the random direction sensitivity of memorized prompts is still substantially elevated compared to non-memorized prompts on SD v2.0, confirming that the sensitivity increase is not confined to the Δ direction alone but spread broadly across the condition embedding space. This supports the use of the direction-agnostic Frobenius norm as a detection feature on both models.

Second, Fig. 6b and Fig. 6d directly compare $\|J_c\|_F$ with $\|J_c \cdot \hat{\Delta}\|_2$. The dashed line in each plot indicates the expected relationship under the assumption that the Jacobian acts isotropically. Under this assumption, the directional sensitivity along any unit vector $\hat{u}$ satisfies

$$\|J_c \cdot \hat{u}\|_2 = \frac{\|J_c\|_F}{\sqrt{d}},$$

where d is the dimension of the condition embedding space. On SD v1.4, memorized prompts cluster near the isotropic baseline, indicating that the sensitivity is distributed relatively evenly across

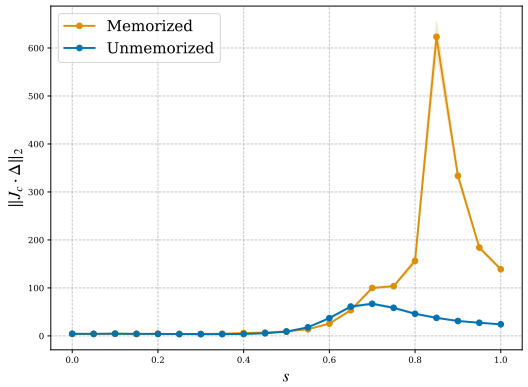
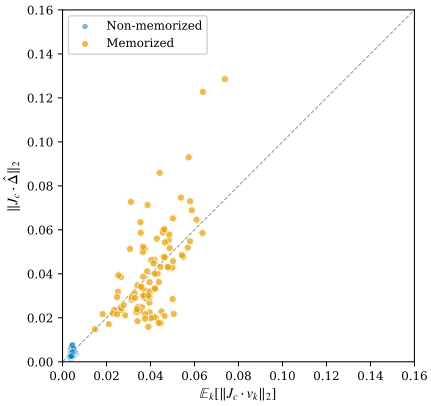
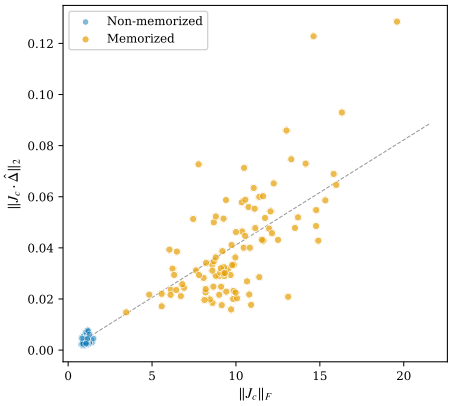


Figure 5: Per-segment contribution $\|J_c(x_t, t, c_{\lambda_i}) \cdot \Delta\|_2$ along the interpolation path from $c_\emptyset$ ($\lambda = 0$) to c ($\lambda = 1$) on SD v2.0. The same near-endpoint localization observed on SD v1.4 (Fig. 2 in the main text) is reproduced here.

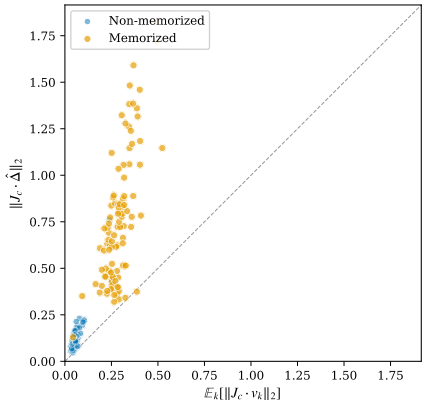
directions, and $\|J_c\|_F$ captures the directional sensitivity $\|J_c \cdot \hat{\Delta}\|_2$ well through the isotropic scaling alone. On SD v2.0, memorized prompts lie noticeably above the isotropic baseline, reflecting the stronger concentration along the Δ direction observed in Fig. 6a and Fig. 6c. Despite this deviation from isotropy, the key observation is that $\|J_c \cdot \hat{\Delta}\|_2$ increases monotonically with $\|J_c\|_F$ on both models: prompts with larger Frobenius norms consistently exhibit larger directional sensitivities. This monotonic relationship ensures that $\|J_c\|_F$ preserves the ranking between memorized and non-memorized samples, the Frobenius norm clearly separates the two classes on both SD v1.4 and SD v2.0. This confirms that $\|J_c\|_F$ serves as an effective proxy for the directional sensitivity without requiring knowledge of the specific direction Δ.



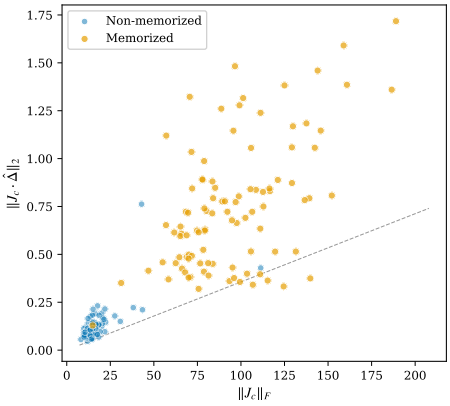
(a) SD v1.4, $\hat{\Delta}$ direction vs. average random direction sensitivity.



(b) SD v1.4, $\|J_c\|_F$ vs. $\|J_c \cdot \hat{\Delta}\|_2$.



(c) SD v2.0, $\hat{\Delta}$ direction vs. average random direction sensitivity.



(d) SD v2.0, $\|J_c\|_F$ vs. $\|J_c \cdot \hat{\Delta}\|_2$.

Figure 6: Directional vs. overall condition sensitivity. Left column: $\|J_c \cdot \hat{\Delta}\|_2$ against the average of $\|J_c \cdot v_k\|_2$ over 20 random unit vectors. Right column: $\|J_c \cdot \hat{\Delta}\|_2$ against $\|J_c\|_F$; the dashed line shows the isotropic baseline $\|J_c\|_F/\sqrt{d}$. Top row: SD v1.4. Bottom row: SD v2.0.