Memorization Detection in Diffusion Models via Text Embedding Interpolation

Changsu Shin [1] **Jinseong Park** [2] **Sungyoon Lee** [3]

Abstract

Text-to-image diffusion models exhibit memorization on certain prompts where the model reproduces a training image. A widely adopted detection approach quantifies the difference between the conditional and unconditional scores. We analyze this difference as a line integral of the conditional score along the linear interpolation from the unconditional embedding to the prompt embedding. Along this interpolation the score trajectory evolves smoothly over most of the interpolation segment and rises sharply within a narrow interval near the prompt embedding, producing the anomalously large score difference. We measure this sharp rise through the condition Jacobian evaluated at the prompt embedding, which achieves state-of-the-art detection accuracy and remains stable across reduced latent resolutions, enabling memory efficient detection by lowering the latent resolution at inference time.

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 approach the score difference as a line integral in 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 segments and rises sharply within a narrow interval near the prompt embedding, forming an abrupt transition that accounts for the anomalously large score difference. The same transition governs the generated image. Below the interval, the generated image follows the same behavior as on a non-memorized interpolation and 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.

Reliable detection therefore reduces to capturing the contribution of this transition, which we measure through the Frobenius norm of the condition Jacobian evaluated at the first denoising step. This signal achieves state-of-the-art detection accuracy and remains stable across initial noises. The same stability extends to the latent resolution, with detection accuracy preserved as the latent resolution is reduced at inference time. Lowering the latent resolution then reduces memory consumption by over two orders of magnitude with minimal accuracy loss, enabling memory efficient detection at a favorable accuracy-memory trade-off.

Our contributions are summarized as follows.

[1]Department of Artificial Intelligence, Hanyang University, Seoul, Republic of Korea [2]Korea Institute for Advanced Study, Seoul, Republic of Korea [3]Department of Computer Science, Hanyang University, Seoul, Republic of Korea. Correspondence to: Jinseong Park <jinseong@kias.re.kr>, Sungyoon Lee <sungyoonlee@hanyang.ac.kr>.

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

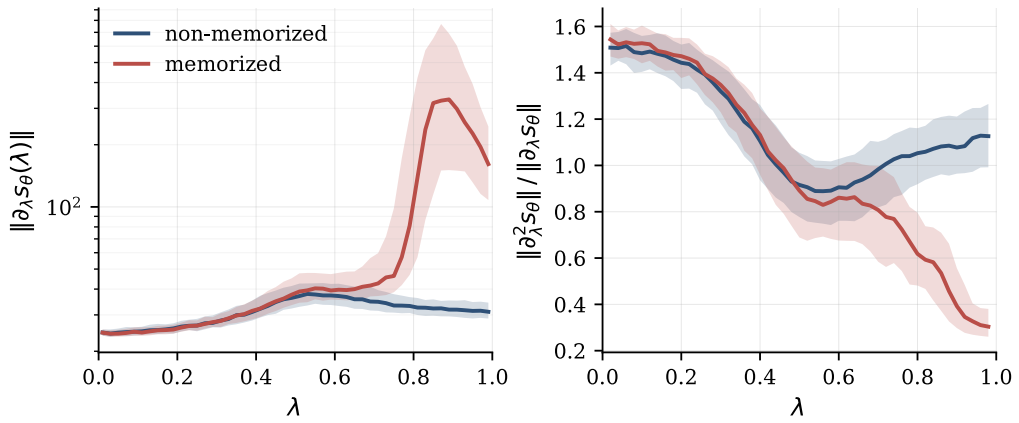


Figure 1. Score variation along the interpolation c_λ at $t = T$. First-order $\log \|\partial s_\theta(\lambda)\|$ and normalized second-order $\|\partial^2 s_\theta\|/\|\partial s_\theta\|$ of s_θ along λ at $t = T$ approximated by finite differences and measured over 100 memorized and 100 non-memorized 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.

- We interpret the score difference as a line integral over text embedding space and show that the score variation does not vary gradually along the score trajectory but arises abruptly 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 and reduced latent resolutions, enabling memory efficient detection.
- 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 accordingly divide 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 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) perform one-step generation across multiple seeds and detect memorization through edge consistency among the outputs. Hintersdorf et al. (2024) computes pairwise SSIM between outputs generated from different seeds. Ren et al. (2024) measures the entropy of cross-attention scores.

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 reinterpret d_t as 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

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

The magnitude d_t is therefore determined by $r_\lambda \triangleq \|\partial_\lambda s_\theta(x_t, t, c_\lambda)\|_2$ along λ which behaves differently between memorized and non-memorized prompts.

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 segments (He et al., 2024; Deckers et al., 2024). As shown in Figure 1, the score trajectory along the linear interpolation from an unconditional embedding c_ϕ to a non-memorized prompt embedding c evolves gradually in λ with low curvature, suggesting that the function the model has learned on intermediate embeddings between c_ϕ and an ordinary prompt varies smoothly along this path. As a result, the rate of change r_λ remains comparable across $\lambda \in [0, 1]$, so that s_{c_λ} varies gradually as λ changes, without a jump at a particular value of λ.

Memorized prompts. The smoothness observed in Figure 1 suggests a prediction for memorized prompts. Memorization at high noise requires d_t to be anomalously large, and if c_{mem} were governed by the same smooth function as ordinary embeddings, this anomalously large separation could only arise from a uniformly large r_s along the entire interpolation segments. Figure 1 shows that this is not the case. For much of the segments, the trajectory is essentially indistinguishable from that of a non-memorized pair, with r_s stays within the typical range. The two curves only separate near the memorized end of the segments, typically around $\lambda \approx 0.7$ in our experiments, beyond which r_λ grows abruptly and most of the change in d_t occurs within a narrow neighborhood of c_{mem}.

The response near c_{mem} is therefore a local discontinuity rather than a continuation of the smooth map governing ordinary regions of text embedding space, and this discontinuity is what makes d_t large for memorized prompts. Precisely measuring d_t thus reduces to accurately capturing the contribution of this discontinuity.

3.2. Detection via the Condition Jacobian

The rate of change r_s analyzed in Section 3.1 can be written via the chain rule as

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

where

$$J_c(x_t, t, c) \;\triangleq\; \frac{\partial s_\theta(x_t, t, c)}{\partial c}. \tag{4}$$

is the condition Jacobian. The abrupt shift identified in Section 3.1 therefore manifests as a sharp rise of $\|J_c(x_t, t, c_s)\,\Delta\|_2$ near c_{mem}, and precise measurement of d_t reduces to capturing this rise.

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

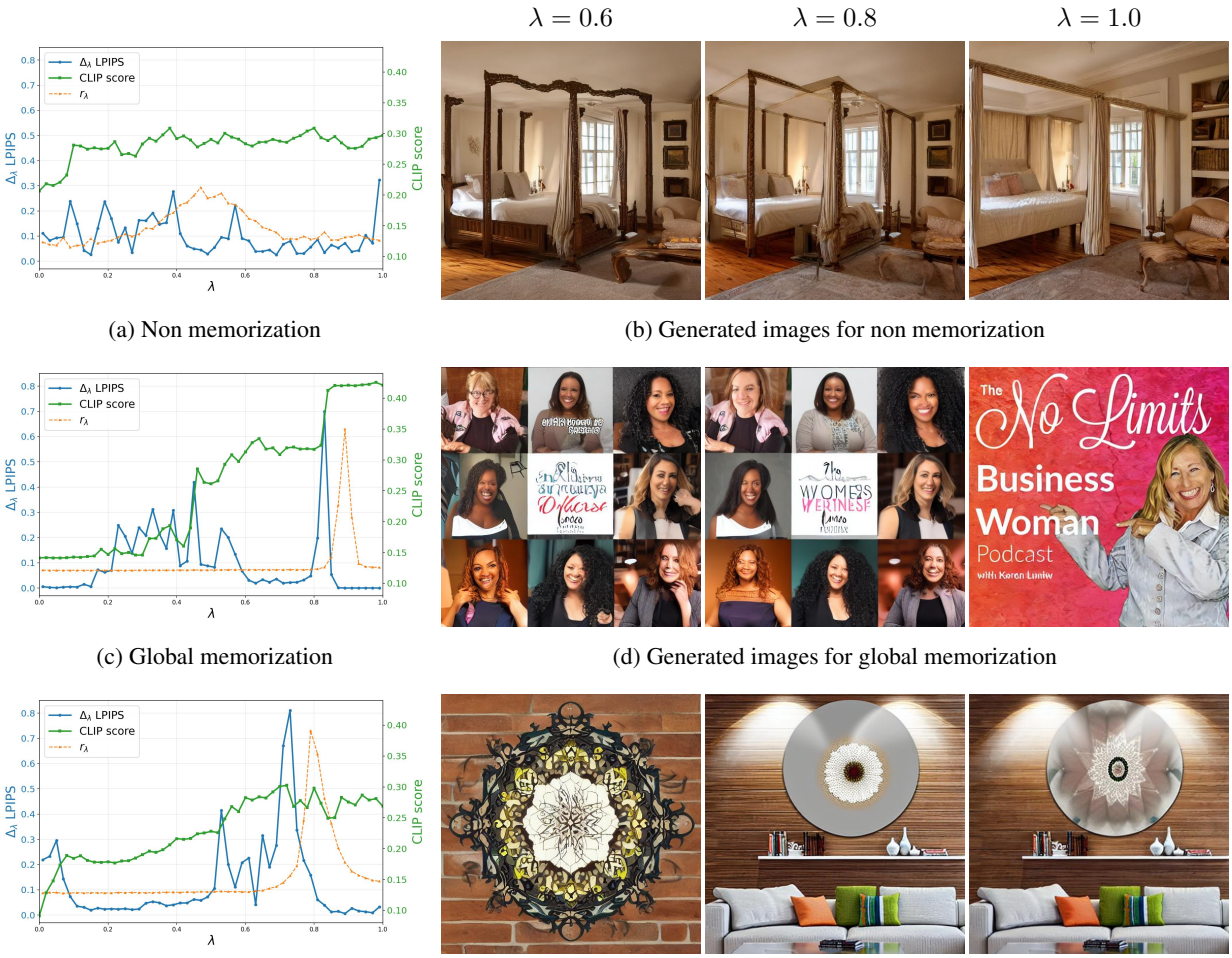
$$\|J_c\,\Delta\|_2 \;\le\; \|J_c\|_F\,\|\Delta\|_2, \tag{5}$$

so we summarize J_c by its Frobenius norm

$$n_c(c) \;\triangleq\; \|J_c(x_T, T, c)\|_F. \tag{6}$$

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 along all 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 E.

Complementary axis from input-noise sensitivity. The same VJP-based estimator also yields the sensitivity of s_θ to x_T at the same forward pass. We define the input-noise



(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.

Jacobian and its norm as

$$J_x(x_T, T, c) \triangleq \frac{\partial s_\theta(x_T, T, c)}{\partial x_T}, \tag{7}$$

$$n_x(c) \triangleq \|J_x(x_T, T, c)\|_F. \tag{8}$$

A single backward pass evaluated at (x_T, c) propagates VJPs to both x_T and c simultaneously, so n_x and n_c can be estimated jointly without any additional forward evaluation, and adding n_x incurs negligible overhead once n_c is being computed.

J_x coincides up to a scalar factor with the Hessian of the log-probability density used by Jeon et al. (Jeon et al., 2025) as their memorization signal, since $\nabla_{x_T} \log p_\theta(x_T) \propto s_\theta(x_T, T, c)$. They report that at the early stage the Hessian magnitudes of memorized and non-memorized prompts are close to each other and difficult to distinguish at this level alone, which forces them to amplify the signal through more refined and costly computations. In our setting, n_x does not need to provide separation on its own since it is combined with n_c, which already separates memorized from non-memorized prompts on its own. The joint feature (n_x, n_c) enlarges the separation margin beyond what either axis achieves alone, so a small number of Hutchinson iterations on n_x suffices for reliable improvement over n_c alone.

Joint feature. We use both quantities together,

$$\phi(c) \triangleq \big(n_x(c),\, n_c(c)\big), \tag{9}$$

and classify by a linear decision rule

$$g(n_x, n_c) = n_c - (a\, n_x + b), \tag{10}$$

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. A sharp change in s_θ implies a corresponding sharp change in the image $x_0(c_\lambda)$ generated under c_λ across this interval, so the same transition appears in image space. 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.

- r_λ from Section 3.1 evaluated at $t = T$.
- $L_i \triangleq \mathrm{LPIPS}(x_0(c_{\lambda_{i+1}}), x_0(c_{\lambda_i}))$.
- $C_i \triangleq \mathrm{CLIP}(x_0(c_{\lambda_i}), c)$.

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

$$\lambda^* \triangleq \arg\max_{\lambda \in [0,1]} r_\lambda. \tag{11}$$

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. 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 training image, which is then reproduced for all larger λ. The two memorization types diverge in their 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.

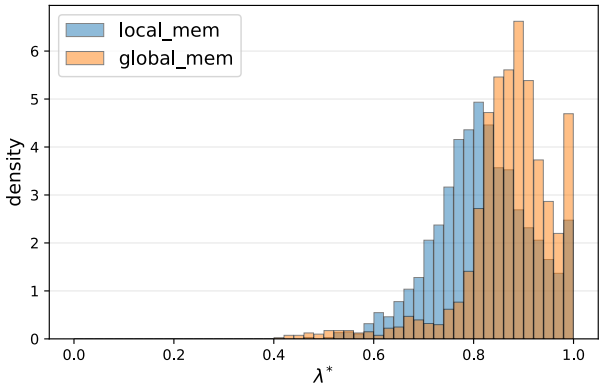


Figure 3. Distribution of λ^* on SD v1.4 over locally and globally memorized prompts. Local memorization yields smaller λ^*.

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 on 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 embedding 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 three 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).

Table 1. 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)	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.995 \pm 0.002}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.995 \pm 0.004}$

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 (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 1 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 $n = 1$.

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

Table 2. 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	Wen	Hintersdorf	Jeon	Asthana	**Ours**
$r = 8$	0.567	**0.884**	0.559	0.821	0.828	0.743
$r = 16$	0.719	**0.951**	0.401	0.935	0.893	0.895
$r = 32$	0.837	0.779	0.514	0.691	0.627	**0.946**
$r = 64$	0.903	0.986	0.876	0.995	0.986	**0.999**
TPR@1%FPR						
$r = 8$	0.002	**0.489**	0.125	0.349	0.167	0.215
$r = 16$	0.011	**0.652**	0.068	0.546	0.441	0.494
$r = 32$	0.054	0.238	0.155	0.045	0.106	**0.574**
$r = 64$	0.248	0.892	0.478	0.968	0.891	**0.995**

SD v2.0						
AUC	Ren	Wen	Hintersdorf	Jeon	Asthana	**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	0.969	0.976	0.968	**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	0.818	0.821	0.824	0.795	**0.996**

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 4 and Table 5 in Appendix B.

5.2. Efficient Detection

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 2 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. Across both models, existing methods exhibit a substantial drop in detection performance as the resolution decreases.

On SD v2.0, the joint feature retains its discriminative power across all tested resolutions. Our method maintains an AUC above 0.987 from the default resolution $4 \times 96 \times 96$ down to $4 \times 8 \times 8$ and a TPR@1%FPR above 0.908 over the same range. At the smallest tested resolution $4 \times 8 \times 8$, our method already exceeds the performance every baseline achieves at the default resolution $4 \times 96 \times 96$.

On SD v1.4, all methods degrade at resolutions below $4 \times$

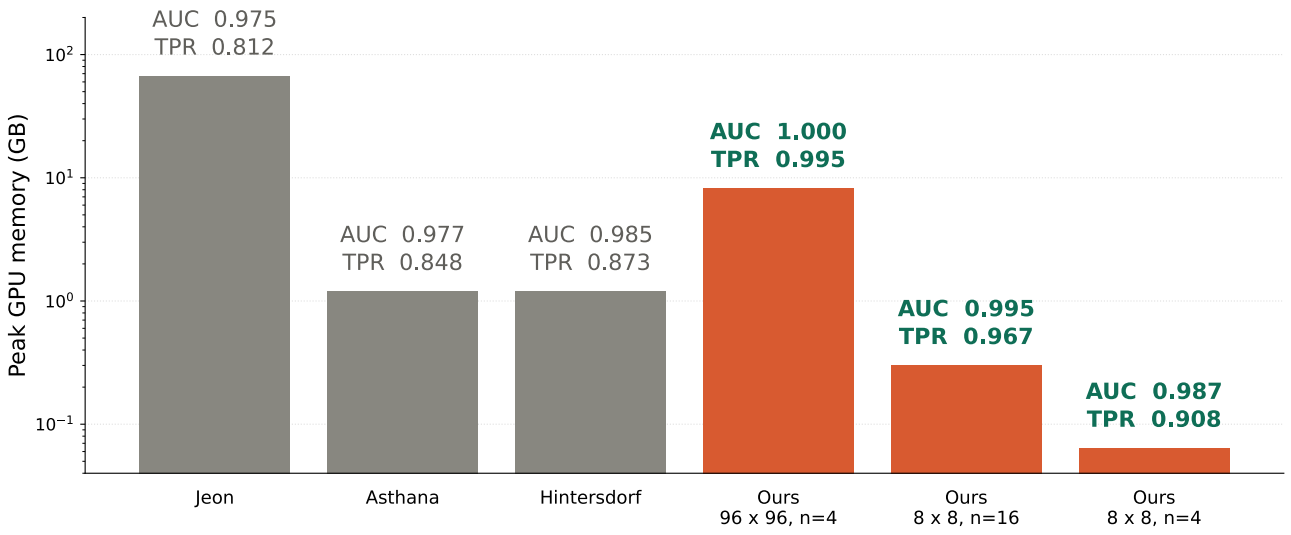


Figure 4. **Memory efficient detection on SD v2.0.** Each method is shown as a point in the (AUC, TPR@1%FPR) plane. At the default $4\times96\times96$ resolution with $n = 4$, baselines (Ren et al., 2024; Wen et al., 2024; Hintersdorf et al., 2024; Asthana & Belagiannis, 2026) consume $\sim$1.2 GB while our method requires $\sim$8.2 GB and Jeon et al. (2025) consumes $\sim$67 GB. Reducing the latent resolution to $4\times8\times8$ lowers our memory to $\sim$64 MB with $n = 4$ at performance comparable to the best baseline at the default resolution, and to $\sim$300 MB with $n = 16$ surpassing all baselines on both AUC and TPR@1%FPR.

32×32. Since the degradation appears across all detection methods, its source lies in the denoiser s_θ on this model rather than in the choice of detection method. SD v2.0 supports resolution reduction down to $4 \times 8 \times 8$ without comparable degradation, and we apply memory efficient detection on this model.

Memory Efficient Detection. Figure 4 reports the memory-accuracy trade-off on SD v2.0. The best baseline Hintersdorf et al. (2024) consumes $\sim$ 1.2 GB at AUC $= 0.985$ and TPR@1%FPR $= 0.873$ with $n = 4$ at the default resolution $4 \times 96 \times 96$. Reducing the latent resolution to $4 \times 8 \times 8$ with $n = 4$ lowers the peak GPU memory of our method to $\sim$ 64 MB, $19\times$ below the best baseline, at AUC $= 0.987$ and TPR@1%FPR $= 0.908$. Increasing the sample count to $n = 16$ raises the memory to $\sim$ 300 MB, still $4\times$ below the best baseline, and the performance to AUC $= 0.995$ and TPR@1%FPR $= 0.967$, exceeding every baseline at the default resolution on both metrics.

6. Limitation

Our method attains state-of-the-art memorization detection accuracy and retains this accuracy under reduced latent resolutions enabling memory efficient detection at a favorable accuracy-memory trade-off. 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.

References

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

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In Calandrino, J. A. and Troncoso, C. (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, C., Liu, D., Shah, M., and Xu, C. Exploring local memorization in diffusion models via bright ending attention. In Yue, Y., Garg, A., Peng, N., Sha, F., and Yu, R. (eds.), *International Conference on Learning Representations*, volume 2025, pp. 29827–29841, 2025.

Deckers, N., Peters, J., and Potthast, M. 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.

He, Q., Wang, J., Liu, Z., and Yao, A. Aid: Attention interpolation of text-to-image diffusion. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 97766–97799. Curran Associates, Inc., 2024. doi: 10.52202/079017-3101.

Hintersdorf, D., Struppek, L., Kersting, K., Dziedzic, A., and Boenisch, F. Finding nemo: Localizing neurons responsible for memorization in diffusion models. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 88236–88278. Curran Associates, Inc., 2024. doi: 10.52202/079017-2800.

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

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

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

Jain, A., Kobayashi, Y., Shibuya, T., Takida, Y., Memon, N., Togelius, J., and Mitsufuji, Y. 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.

Jeon, D., Kim, D., and No, A. 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.

Jin, C., Shi, Q., and Gu, Y. 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.

Karris, N., Durell, L., Flores, J., and Emerson, T. Which way from b to a: The role of embedding geometry in image interpolation for stable diffusion. *arXiv preprint arXiv:2511.12757*, 2025.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T. (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125.

Ren, J., Li, Y., Zeng, S., Xu, H., Lyu, L., Xing, Y., and Tang, J. Unveiling and mitigating memorization in text-to-image diffusion models through cross attention. In Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., and Varol, G. (eds.), *Computer Vision – ECCV 2024*, pp. 340–356, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-72980-5.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. 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.

Shen, X., Qu, Y., Backes, M., and Zhang, Y. Prompt Stealing Attacks Against Text-to-Image Generation Models. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2024.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. Understanding and mitigating copying in diffusion models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47783–47803. Curran Associates, Inc., 2023a.

Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. 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.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

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

Wen, Y., Liu, Y., Chen, C., and Lyu, L. 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 3. 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
$K = 4$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.995 \pm 0.002}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.995 \pm 0.004}$

A. 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 3 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.

B. 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 4 and 5 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 (Jeon et al.,

Table 4. 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	$\mathbf{0.976 \pm 0.005}$	$\mathbf{0.977 \pm 0.008}$	$\mathbf{0.988 \pm 0.007}$	$\mathbf{0.993 \pm 0.007}$

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 (Asthana & 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.

C. 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. 5 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.

D. Detection on Another Model

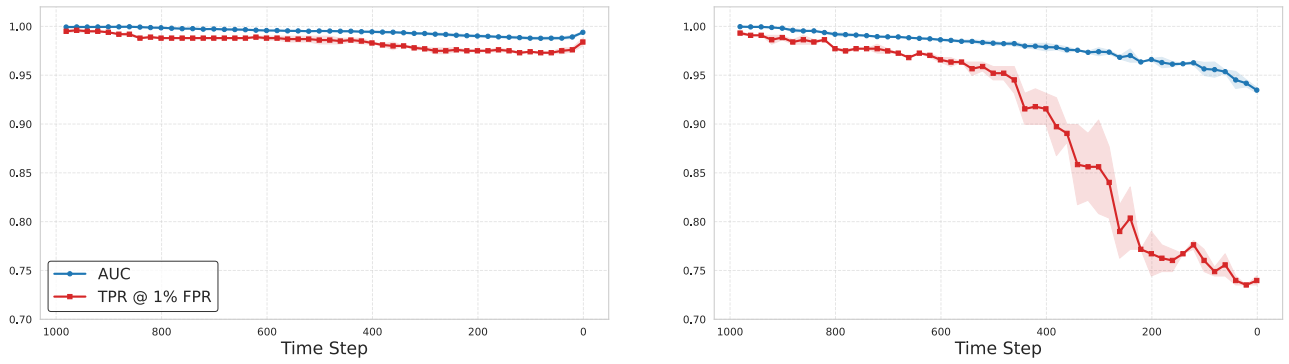
To evaluate whether our joint detection method generalizes beyond the Stable Diffusion models, we conduct additional experiments on Realistic Vision v5.1[1], a community fine-tuned model based on Stable Diffusion v1.5. We use the memorized

[1]https://civitai.com/

Table 5. 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	$\mathbf{0.944 \pm 0.023}$	$\mathbf{0.958 \pm 0.010}$	$\mathbf{0.967 \pm 0.026}$	$\mathbf{0.980 \pm 0.012}$



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

Figure 5. 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.

prompt set provided by Webster (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 6 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.

E. 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

Table 6. Memorization detection performance on Realistic Vision v5.1. Mean ± 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

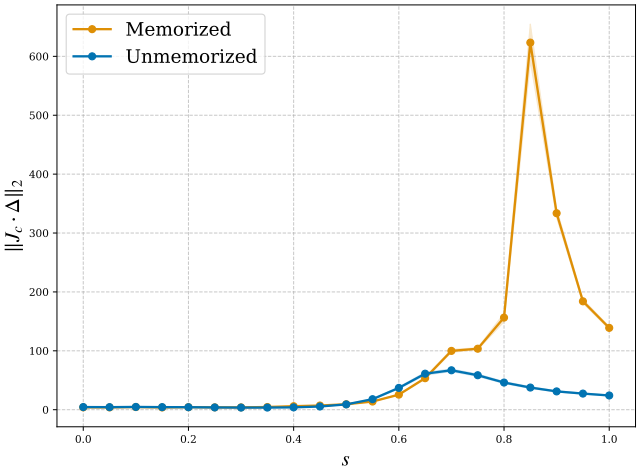


Figure 6. Per-segment contribution $\|J_c(x_t, t, c(s_i)) \cdot \Delta\|_2$ along the interpolation path from $c_\emptyset$ ($s = 0$) to c ($s = 1$) on SD v2.0. The same near-endpoint localization observed on SD v1.4 (Fig. 2 in the main text) is reproduced here.

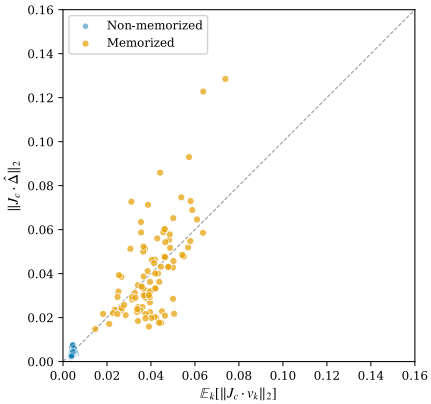
the $\Delta = c - c_\emptyset$ direction, in which case $\|J_c\|_F$ would be a loose upper bound and a poor proxy. To investigate this, we examine two comparisons on both SD v1.4 and SD v2.0.

First, Fig. 7a and Fig. 7c 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.

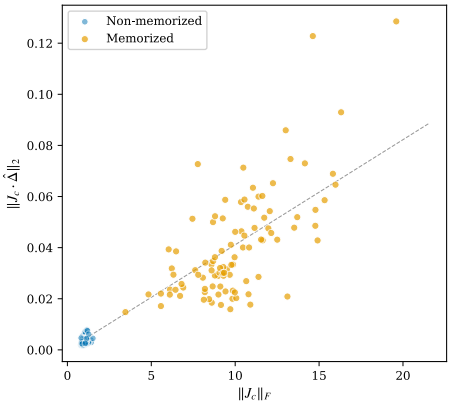
Second, Fig. 7b and Fig. 7d directly compare $\|J_c\|_F$ with $\|J_c \cdot \hat{\Delta}\|_2$. The dashed line in each plot indicates the expected relationship under the assumption that the Jacobian acts isotropically. Under this assumption, the directional sensitivity along any unit vector $\hat{u}$ satisfies

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

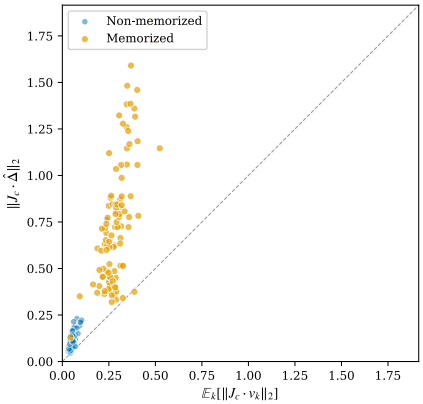
where d is the dimension of the condition embedding space. On SD v1.4, memorized prompts cluster near the isotropic baseline, indicating that the sensitivity is distributed relatively evenly across directions, and $\|J_c\|_F$ captures the directional sensitivity $\|J_c \cdot \hat{\Delta}\|_2$ well through 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. 7a and Fig. 7c. 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.



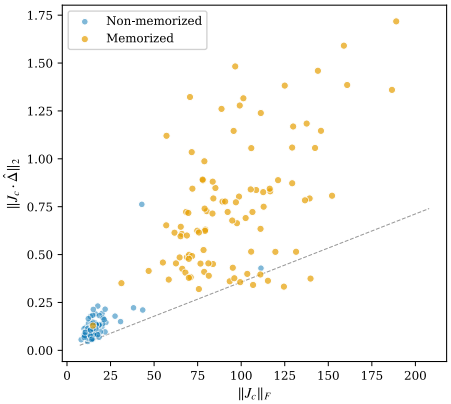
(a) SD v1.4, $\hat{\Delta}$ direction vs. average random direction sensitivity.



(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 7. 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.