PATCHRECOVER: IMAGE-CENTRIC DIFFUSION MEMORIZATION DETECTION VIA LOCAL COUNTERFACTUAL PATCH RECOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models can memorize parts of training images, posing privacy and copyright risks. Many existing detectors rely on global, prompt-conditioned signals, which can obscure local memorization and limit detection when only generated images are available. We propose *PatchRecover*, a framework that extracts local evidence through counterfactual patch recovery and can operate without the original prompt, seed, or denoising trajectory. The core idea is that memorized content should recover more readily after a local edit than non-memorized content. We test this by replacing a patch with a counterfactual alternative, regenerating the image, and measuring recovery toward the original content. We aggregate the strongest patch-level recovery margins to obtain an image-level detection score. To explain why memorized patches recover more strongly, we introduce local recovery regions and derive a reconstruction bound under a local contraction assumption. Trajectory measurements show that memorized patches enter these regions more often and recover more strongly than non-memorized patches. On Stable Diffusion v1.4 and v2-base benchmarks that include non-memorized outputs from memorization-associated prompts, PatchRecover achieves mean image-level AUCs of 0.9980 and 0.9968, respectively. When both memorized and non-memorized images come from memorization-associated prompts, PatchRecover improves mean AUC from the strongest baseline's 0.9374 to 0.9782.

1 INTRODUCTION

Text-to-image diffusion models such as Stable Diffusion (Rombach et al., 2022) can reproduce training content instead of synthesizing novel images (Somepalli et al., 2023a; Carlini et al., 2023; Webster, 2023), creating privacy and copyright risks. Memorization need not cover an entire image. A model may repeatedly recover a particular object, logo, or texture while the surrounding content varies. A practical detector should therefore determine both whether an image contains memorized content and which regions provide that evidence.

Existing detectors measure conditional–unconditional noise-prediction differences (Wen et al., 2024), cross-attention concentration (Ren et al., 2024), probability-landscape sharpness (Jeon et al., 2025), or score anisotropy (Asthana & Belagiannis, 2026). Aggregating such evidence over an entire image can obscure memorization confined to a small region. End-token attention (Chen et al., 2025) and coordinate-wise curvature (Kim & Lee, 2026) provide spatial evidence, but generally use prompt-conditioned model queries. Per-image optimization of a noise distribution and conditioning yields a single image-level memorization score rather than localized evidence (Ma et al., 2025). The remaining question is how to obtain local memorization evidence directly from an image without requiring its original prompt, seed, or denoising trajectory, while avoiding per-image optimization.

We address this question with *PatchRecover*, a framework that detects memorization through local counterfactual patch recovery. The core idea is that, after a local edit, memorized content tends to return toward its original form more strongly than non-memorized content. As shown in Figure 1, we replace a patch with a counterfactual alternative and regenerate both the edited and original images using the same randomness and sampling settings. When the prompt is unavailable, the prompt-free variant, *PatchRecover-PF*, uses empty text conditioning. The no-edit output provides a reference

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.

for changes caused by regeneration itself. A larger recovery margin means that the edited output is closer to this reference than to the counterfactual patch. We aggregate the strongest patch scores into an image-level decision, without specifying the memorized region in advance.

Our analysis introduces the concept of a *local recovery region*: a neighborhood in which regeneration contracts toward the original content in expectation. Under a local contraction assumption, we bound the final reconstruction error. We then empirically analyze regeneration trajectories of memorized and non-memorized patches on both models. Memorized patches enter recovery regions more often and recover more strongly, including within the same entry group. These observations support the recovery-based explanation of the detection score.

We evaluate image-level detection on Stable Diffusion v1.4 and v2-base, including non-memorized images generated from memorization-associated prompts. This setting tests whether a detector responds to the image's content rather than just its prompt's memorization tendency. PatchRecover achieves the highest mean image AUC on both benchmarks and leads AUC and TPR in the targeted SD1 mismatch comparisons. Together, the results demonstrate the value of local image evidence for memorization detection, while the recovery analysis gives a concrete interpretation of that evidence.

Contributions.

1. We propose *PatchRecover*, an image-centric framework for diffusion memorization detection based on local counterfactual patch recovery. PatchRecover operates directly on an observed generated image, requires neither the source seed nor the source denoising trajectory, and supports both prompt-free and prompt-available settings.
2. We provide a contraction-based reconstruction bound for local patch recovery and analyze trajectory measurements on both models. Memorized patches enter recovery regions more often and show stronger final recovery, including within the same entry group.
3. We curate memorization detection benchmarks for Stable Diffusion v1.4 and v2-base that include non-memorized images from memorization-associated prompts. A targeted mismatch test assesses whether detectors distinguish these images from memorized outputs, rather than relying on prompt association alone.
4. We compare PatchRecover with six memorization detectors on image-level detection and exploratory patch-level evaluations. Ablations examine the effects of local edits, distance measures, regeneration randomness, and patch aggregation.

2 Related Work

Diffusion models and training-data replication. Diffusion models generate images by reversing a gradual noising process (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b). Stable Diffusion runs this process in a learned latent space (Rombach et al., 2022). DDIM (Song et al., 2021a) and image regeneration via noise addition and denoising (Meng et al., 2022) provide the sampling tools underlying our method. Beyond generating novel images, diffusion models can replicate training content. Prior work examines this in several ways: Somepalli et al. (2023a) study replication through retrieval, Carlini et al. (2023) extract training examples through generation and filtering, and Webster (2023) demonstrates reproducible extraction across text-to-image systems. Membership inference asks a distinct question: whether a candidate sample was used in training (Duan et al., 2023; Matsumoto et al., 2023), rather than whether a generated image reproduces training content. We focus on the latter. Within this setting, text conditioning influences copying (Somepalli et al., 2023b), but a memorization-associated prompt need not produce memorized content in every generation. We therefore distinguish a prompt's memorization tendency from the memorization status of an individual output image.

Memorization detection methods. Model-response detectors use conditional–unconditional noise-prediction differences (Wen et al., 2024), cross-attention concentration (Ren et al., 2024), or end-token attention (Chen et al., 2025). Geometric approaches examine sharpness (Jeon et al., 2025), anisotropy (Asthana & Belagiannis, 2026), and coordinate-wise curvature differences (Kim & Lee, 2026), while Broken Memories (Huang et al., 2026) detects anomalous latent updates during generation. Other detectors instead optimize noise or conditioning for each candidate image.

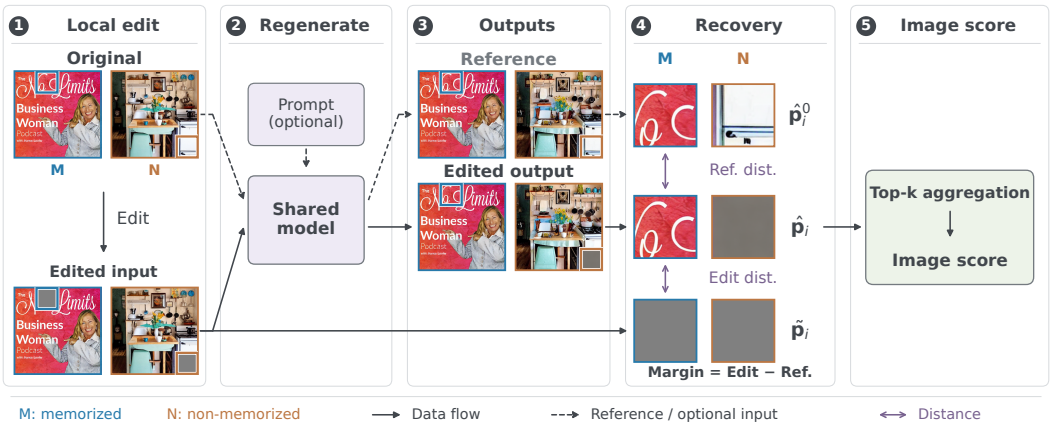

Figure 1: PatchRecover pipeline. Matched regenerations yield patch recovery margins, aggregated into an image score. Dashed paths denote the no-edit reference. The prompt input is optional.

IIP (Jiang et al., 2025) pairs unconditional DDIM inversion with optimized prompt perturbations, scoring memorization by latent reconstruction or noise-prediction discrepancy. InvMM (Ma et al., 2025) optimizes an image-specific noise distribution and soft prompt conditioning, scoring memorization by KL divergence from the Gaussian prior subject to reproducing the image. Unlike these per-image optimization approaches, PatchRecover measures responses to local edits under matched regeneration, aggregating recovery margins into an image-level score. Besides the input image, it needs only access to the generating model. It requires neither the source seed nor the denoising trajectory, and prompt conditioning is optional.

3 Method

3.1 PatchRecover: Local Counterfactual Patch Recovery

Let $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$ be an image generated by a diffusion model G_θ under prompt or conditioning c, divided into n non-overlapping patches $\boldsymbol{x} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$. Let $\boldsymbol{x}_{\setminus i}$ denote the image context outside patch i. We consider a family of local patch interventions $\mathcal{T}$ (e.g., blank gray, neighbor-patch replacement, flipping, JPEG degradation, and blur) and, for simplicity, fix one $T \in \mathcal{T}$ throughout our analysis. Under T, the counterfactual patch is $\tilde{\boldsymbol{p}}_i = T(\boldsymbol{p}_i)$ and the counterfactually modified image is $\boldsymbol{x}_i = \boldsymbol{x}_{\setminus i} \cup \tilde{\boldsymbol{p}}_i$. "Counterfactual" here means that the global context $\boldsymbol{x}_{\setminus i}$, prompt c, model G_θ, and patch location are held fixed, and only the local patch is replaced by $T(\boldsymbol{p}_i)$.

We then regenerate the modified image with an image-to-image pipeline,

$$\hat{\boldsymbol{x}}_i = R_\theta(\boldsymbol{x}_i, c, \xi), \qquad \xi \sim P_\xi,$$

where R_θ is the regeneration procedure and ξ is the randomness, including the noise that initializes the noised latent. The condition c can be set to empty in the prompt-free setting. The regenerated patch at location i is $\hat{\boldsymbol{p}}_i = \hat{\boldsymbol{x}}_i[i]$. To control for ordinary regeneration variation, we also regenerate the original image under the same randomness as a reference branch: $\hat{\boldsymbol{x}}^0 = R_\theta(\boldsymbol{x}, c, \xi)$ and $\hat{\boldsymbol{p}}_i^0 = \hat{\boldsymbol{x}}^0[i]$. Let $d(\boldsymbol{a}, \boldsymbol{b})$ be a generic patch dissimilarity, where smaller values indicate stronger visual similarity. In practice, d is the L2 distance or a similarity measure converted into a dissimilarity. Figure 1 illustrates the PatchRecover pipeline. We locally edit each patch and regenerate both the original and edited images using the same model. Memorization is measured by how strongly the edited patch is recovered toward its reference regeneration. We define the practical recovery margin:

$$M_T(\boldsymbol{p}_i) = d(\hat{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}_i) - d(\hat{\boldsymbol{p}}_i, \hat{\boldsymbol{p}}_i^0), \tag{1}$$

where a larger margin indicates stronger recovery toward the reference. To obtain the image-level score, we rank the patch margins as $M_T(\boldsymbol{p}(1)) \geq \cdots \geq M_T(\boldsymbol{p}(n))$ and average the top-k margins:

$$S_k(\boldsymbol{x}) = \frac{1}{k}\sum_{j=1}^{k} M_T(\boldsymbol{p}_{(j)}). \tag{2}$$

We use $k = 2$ by default, while retaining all patch-level margins $\{M_T(\boldsymbol{p}i)\}_{i=1}^n$ to localize memorized regions. Algorithm 1 in Appendix C summarizes the full inference procedure.

3.2 Local Recovery Region

We now study the mechanism underlying the counterfactual recovery exploited by PatchRecover. The practical margin M_T measures this recovery through paired regenerations. For analysis, we first characterize the recovery outcome more directly by comparing the regenerated patch with the original patch and its counterfactual. Using the patch dissimilarity d from Section 3.1, we define the **patch-level recovery dissimilarity gap** as

$$D_T(\boldsymbol{p}_i) = d(\hat{\boldsymbol{p}}_i, \boldsymbol{p}_i) - d(\hat{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}_i). \tag{3}$$

A smaller $D_T(\boldsymbol{p}_i)$ indicates that the regenerated patch $\hat{\boldsymbol{p}}_i$ is closer to the original patch $\boldsymbol{p}_i$ than to its counterfactual $\tilde{\boldsymbol{p}}_i$, and thus stronger recovery of the original content. This quantity is closely connected to the practical margin M_T: when the reference regeneration satisfies $\hat{\boldsymbol{p}}_i^0 \approx \boldsymbol{p}_i$, we have $M_T(\boldsymbol{p}_i) \approx -D_T(\boldsymbol{p}_i)$. Empirically, using the matched reference or the original patch yields similar trends, while using M_T better reduces regeneration-induced variation (Appendix D).

While D_T characterizes the recovery outcome, understanding why memorized and non-memorized patches behave differently requires examining the intermediate denoising dynamics. Our key hypothesis is that memorized patches are more likely to reach a neighborhood of their original content where subsequent reverse steps consistently pull them closer to the original patch. We call such a neighborhood a *local recovery region*. This provides a mechanism for counterfactual recovery: even after an intervention moves a patch away from its original content, entering this region can drive the remaining trajectory back toward the original patch.

To formalize this idea, let $\boldsymbol{s}_{i,t}$ denote the full reverse-sampling state at step t, including the latent at reverse timestep t and any scheduler state needed for the next update. Let $\boldsymbol{p}_{i,t}$ denote the corresponding intermediate patch, with $\boldsymbol{p}_{i,0} = \hat{\boldsymbol{p}}_i$ as the final regenerated patch. We first consider a neighborhood of the original patch,

$$\mathcal{R}_i(\boldsymbol{p}_i; \rho_i) = \{\boldsymbol{q} : d(\boldsymbol{q}, \boldsymbol{p}_i) \le \rho_i\}, \qquad \rho_i > 0.$$

Proximity alone, however, does not make this neighborhood a recovery region. The defining property is that, once the trajectory enters the region, the subsequent reverse dynamics contract toward $\boldsymbol{p}_i$, up to a bounded residual error. We formalize this property below.

Definition: Local Recovery Dynamics. We say that $\boldsymbol{p}_i$ has a local recovery region $\mathcal{R}_i(\boldsymbol{p}_i; \rho_i)$ if, once inside it, the reverse dynamics contract toward a bounded neighborhood of $\boldsymbol{p}_i$. Formally, there exists a fixed recovery floor $0 \le \epsilon_i < \rho_i$ such that, for each timestep t, there are constants $0 < \lambda_t < 1$ and $\zeta_t \ge 0$ with $\zeta_t \le (1 - \lambda_t)\epsilon_i$ and, for every state with $\boldsymbol{p}_{i,t} \in \mathcal{R}_i(\boldsymbol{p}_i; \rho_i)$,

$$\mathbb{E}[d(\boldsymbol{p}_{i,t-1}, \boldsymbol{p}_i) \mid \boldsymbol{s}_{i,t}] \le \lambda_t \, d(\boldsymbol{p}_{i,t}, \boldsymbol{p}_i) + \zeta_t. \tag{4}$$

Here λ_t is the contraction strength toward $\boldsymbol{p}_i$, and ζ_t is an additive residual for effects the ideal contraction does not explain (regeneration stochasticity, approximation error, and decoding drift), which can keep the recovery error from reaching zero. The condition $\zeta_t \le (1 - \lambda_t)\epsilon_i$ guarantees contraction whenever the current dissimilarity exceeds ϵ_i.

Proposition 1: Counterfactual Patch Recovery under Local Contraction. *Suppose that, after intervention T, the intermediate patch trajectory enters the local recovery region at timestep t_0, i.e., $\boldsymbol{p}_{i,t_0} \in \mathcal{R}_i(\boldsymbol{p}_i; \rho_i)$, and that the contraction condition in Eq. 4 holds at each reverse step $t = t_0, t_0 - 1, \ldots, 1$. Then*

$$\mathbb{E}[d(\boldsymbol{p}_{i,0}, \boldsymbol{p}_i) \mid \boldsymbol{s}_{i,t_0}] \le \epsilon_i + \left(\prod_{t=1}^{t_0} \lambda_t \right) [d(\boldsymbol{p}_{i,t_0}, \boldsymbol{p}_i) - \epsilon_i]. \tag{5}$$

Because $0 < \lambda_t < 1$, the product $\prod_{t=1}^{t_0} \lambda_t$ is below one, so whenever $d(\boldsymbol{p}_{i,t_0}, \boldsymbol{p}_i) > \epsilon_i$ the expected final dissimilarity is strictly smaller than at entry. Sustained local contraction thus drives the modified patch toward the original up to the recovery floor ϵ_i, without requiring exact reconstruction. The proof is given in Appendix B.1.

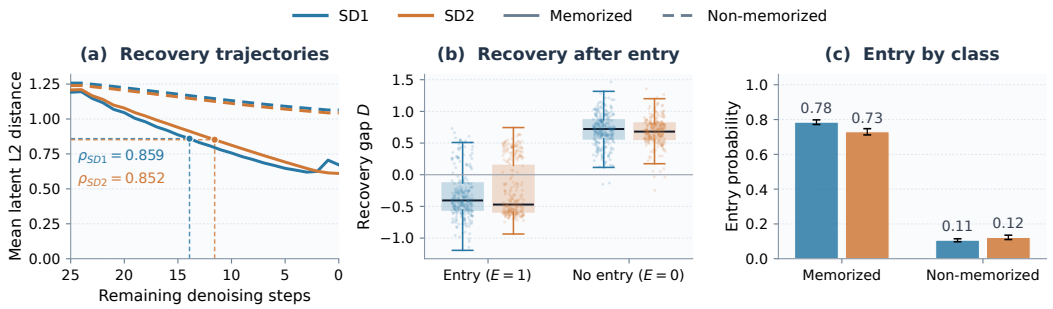

Figure 2: Patch recovery on SD1 and SD2: (a) class-mean trajectories, (b) entry-conditioned recovery gaps, and (c) class-wise entry probabilities. Error bars in (c) are 95% prompt-bootstrap intervals.

Empirical analysis. We examine whether the denoising trajectories of memorized and non-memorized patches exhibit the recovery behavior hypothesized above. We use the setup in Figure 2: gray edits (the queried patch replaced by constant gray), raw latent L2 distance as d, source-prompt conditioning, and the original generation seed. During regeneration, we record the intermediate patch trajectory $\{\boldsymbol{p}_{i,t}\}_{t=0}^{t_{\max}}$ and its distance to the original patch latent, $R_{i,t}^T = d(\boldsymbol{p}_{i,t}, \boldsymbol{p}_i)$, where smaller values indicate stronger recovery. Ground-truth image labels are used only to group the resulting trajectories for analysis, not as inputs to the regeneration or PatchRecover.

For this analysis, we operationalize the recovery region using a single threshold ρ for each model. We sweep the observed values of $R_{i,1}^T$, the step immediately before the final output, and choose the smallest ρ that best separates the observed trajectories of the two groups. This yields $\rho = 0.8594$ for SD1 and $\rho = 0.8519$ for SD2. Details are provided in Appendix B.3. Figure 2(a) shows the trajectories averaged within each ground-truth group. Consistent with our hypothesis, patches from memorized images approach their original content faster and enter the low-distance region, whereas patches from non-memorized images generally remain above ρ. Means are shown only for visualization. Region entry is computed on individual trajectories. The late rebound for SD1 also motivates allowing a nonzero recovery floor rather than assuming exact convergence.

Figure 2(b) then examines whether entering this region corresponds to stronger recovery. Let E_i^T indicate whether trajectory i enters the region at any non-final denoising step. Trajectories with entry ($E_i^T = 1$) have substantially smaller recovery gaps $D_T(\boldsymbol{p}_i)$ than those without ($E_i^T = 0$) on both models. Figure 2(c) further shows that this entry behavior is much more frequent among memorized patches. Together, these observations support the hypothesized local recovery behavior and motivate the separation analysis in Section 3.3.

3.3 Recovery-Region Entry Separability

We now connect local recovery to the separation between patches from memorized images (M, $Y_i = 1$) and patches from non-memorized images (N, $Y_i = 0$). Using the entry indicator E_i^T, define the entry probability $\alpha_y^T = \mathbb{P}(E_i^T = 1 \mid Y_i = y)$ and the conditional mean gap $\mu_{e,y}^T = \mathbb{E}[D_T(\boldsymbol{p}_i) \mid E_i^T = e, Y_i = y]$. By the law of total expectation, the difference between class means is

$$\begin{aligned}\Delta_D^T &= \mathbb{E}[D_T(\boldsymbol{p}_i) \mid Y_i = 1] - \mathbb{E}[D_T(\boldsymbol{p}_i) \mid Y_i = 0] \\ &= \alpha_1^T \mu_{1,1}^T + (1 - \alpha_1^T)\mu_{0,1}^T - \alpha_0^T \mu_{1,0}^T - (1 - \alpha_0^T)\mu_{0,0}^T.\end{aligned} \tag{6}$$

Since smaller D_T indicates stronger recovery, $\Delta_D^T < 0$ is the desired ordering.

Sufficient conditions for separation. Intuitively, $\Delta_D^T < 0$ follows when M patches enter more often, entry is associated with stronger recovery among M patches, and M patches recover more strongly than N patches within each entry group. Formally, it suffices that:

(i) M patches enter more often than N patches: $\alpha_1^T > \alpha_0^T$.

(ii) Entering M patches have smaller gaps than non-entering M patches: $\mu_{1,1}^T < \mu_{0,1}^T$.

(iii) M patches have smaller gaps than N patches within each entry group: $\mu^T_{e,1} < \mu^T_{e,0}$ for $e \in \{0, 1\}$.

Then Eq. 6 gives $\Delta^T_D < 0$: both entry frequency and within-group recovery favor M. The weighted-mean derivation, which does not assume equal conditional means, and the empirical conditional means are in Appendix B.2.

Empirical evidence and implication. The recorded trajectories support the proposed separation mechanism on both SD1 and SD2. As shown in Figure 2(c), memorized patches enter the recovery region substantially more often than non-memorized patches: 0.7848 versus 0.1064 on SD1 and 0.7298 versus 0.1223 on SD2. This ordering remains consistent across all tested thresholds (Appendix B.3). Moreover, within each class, trajectories that enter the region have smaller final recovery gaps, and within each entry group, memorized patches have smaller gaps than non-memorized patches. Together, these effects yield class-mean gap differences of -0.9648 on SD1 and -0.8320 on SD2. Thus, the stronger tendency of memorized patches to enter the recovery region translates into stronger counterfactual recovery at the class level, providing a mechanistic basis for the practical recovery margin in Section 3.1. This result establishes mean separation rather than non-overlapping distributions or a particular detection AUC. The practical detection is evaluated in Section 4.

4 Experiments

4.1 Experimental setup

Models and benchmark. We denote memorized and non-memorized images or patches by M and N. We evaluate Stable Diffusion v1.4 (SD1) (CompVis, 2022) and v2-base (SD2) (Manojb, 2022). Candidate memorized prompts from Webster (2023) are filtered for strong cross-seed reproducibility, retaining 482 of 500 prompts for SD1 and 216 of 219 for SD2. Non-memorized prompts follow Jeon et al. (2025) and come from COCO (Lin et al., 2014), Lexica (Shen et al., 2024), Tuxemon (Tsaban & Paul, 2024), and GPT-4 (Achiam et al., 2023). Each retained memorized prompt contributes four frozen generations and is paired with a non-memorized prompt, yielding balanced cohorts of 3,856 SD1 and 1,728 SD2 images for ablations and threshold fitting. The expanded evaluation additionally includes non-memorized images from memorization-associated prompts and the remaining non-memorized prompts. Memorization labels follow the cross-seed reproducibility criterion detailed in Appendix A.

Baselines. We compare against one paper-primary configuration from each of six recent detectors: MemAttn-D from Ren et al. (2024), the all-step score from Wen et al. (2024), the published sharpness score from Jeon et al. (2025), Bright Ending from Chen et al. (2025), anisotropy from Asthana & Belagiannis (2026), and covariance-based coordinate-wise curvature from Kim & Lee (2026). Their released score definitions and native query schedules are preserved. These baselines use the prompt together with noise predictions, attention maps, or derivatives, whereas PatchRecover uses paired regenerated outputs and its prompt-free variant (PatchRecover-PF) uses empty conditioning. InvMM (Ma et al., 2025) is compared only on a matched 72-image subset because of its computational cost (Appendix C.3). IIP (Jiang et al., 2025) and Broken Memories (Huang et al., 2026) are excluded because their implementations were unavailable.

Metrics and implementation. The primary metrics are the area under the ROC curve (AUC) and the true positive rate at 1% false-positive rate (TPR). Image accuracy (ACC) and group-B specificity measure frozen-threshold transfer. PatchRecover scores each image with the recovery margin in Eq. 1 and top-2 aggregation. Regeneration uses a 4×4 query grid, 50 inference steps, strength 0.5, and guidance 7.5, with paired branches sharing the same randomness. The edit and distance are selected on the evaluated image cohorts, and patch detection uses a separate selection based on mean AUC over the 20 draws. Both are then fixed across draws. This selection is retrospective, but subsequent threshold fitting excludes test prompts. Appendix C specifies the edits, distances, and evaluation protocols. On the expanded cohorts, PatchRecover and PatchRecover-PF report mean$\pm$sample standard deviation across four regeneration seeds (the original generation seed and three source-independent seeds). Baseline scores are deterministic given the prompt and query seed, so each is evaluated once.



Figure 3: SD1 examples of groups A, B, and C. Each group shows four images generated from one prompt with different seeds.

Table 1: Image detection under prompt-image mismatch (AUC and TPR@1%FPR). Mean ± SD over 20 A-prompt draws, with both PatchRecover variants first averaged over four regeneration seeds per draw. Best and second-best are in bold and underlined, respectively.

	A_{18} vs. B		A_{36} vs. $B + C$	
Method	AUC ↑	TPR ↑	AUC ↑	TPR ↑
Ren et al.	0.8577±0.0404	0.3535±0.0960	0.9281±0.0182	0.4240±0.0857
Wen et al.	0.9170±0.0240	0.4368±0.1082	0.9486±0.0118	0.4757±0.0857
Jeon et al.	0.8616±0.0164	0.1236±0.0343	0.9257±0.0081	0.1236±0.0262
Chen et al.	<u>0.9374</u>±0.0265	0.4597±0.1216	<u>0.9531</u>±0.0138	0.4833±0.0866
Asthana and Belagiannis	0.8922±0.0269	0.2764±0.0535	0.9366±0.0106	0.3958±0.0403
Kim and Lee	0.9003±0.0211	0.2451±0.0605	0.9496±0.0136	0.3031±0.0570
PatchRecover	**0.9782**±0.0131	**0.6495**±0.0828	**0.9861**±0.0053	**0.7712**±0.0481
PatchRecover-PF	0.8982±0.0284	<u>0.5040</u>±0.0698	0.9288±0.0199	<u>0.5236</u>±0.0796

4.2 Detection under prompt-image mismatch

A memorization-associated prompt does not always produce a memorized image, creating a prompt-image mismatch. Prompt-conditioned detectors may therefore confound the two. We evaluate whether detectors can instead identify memorization from the generated image itself. Group A contains memorized prompts with memorized images, group B contains memorization-associated prompts with non-memorized images under the benchmark criterion, and group C contains non-memorized prompts with non-memorized images (Figure 3 shows selected examples). We focus on SD1, which has 18 group-B prompts, because SD2 has only 3, too few for a comparably informative evaluation. Each of 20 prespecified random draws selects 36 prompts uniformly without replacement for A and keeps the 18 prompts in each of B and C fixed, with four images per prompt. The nested comparison A_{18} versus B isolates prompt-image mismatch with 72 images per class, and A_{36} versus $B + C$ adds negatives from non-memorized prompts, giving 144 images per class.

PatchRecover leads mean AUC and TPR in both mismatch comparisons (Table 1). On A_{18} versus B, its AUC is 0.9782, above Chen et al.'s 0.9374, and its TPR improves from 0.4597 to 0.6495. PatchRecover-PF, whose signal comes from the image alone, has the second-highest TPR in both comparisons despite its lower AUC. Table 7 in Appendix C.5 further evaluates whether thresholds fitted on the full benchmark transfer to these comparisons, since a deployed detector needs a fixed decision threshold. Each draw freezes thresholds fitted on SD1 with all test prompts excluded. Under this transfer, PatchRecover achieves the highest mean ACC (0.8476 and 0.9124), while PatchRecover-PF and PatchRecover have comparable group-B specificity (0.7097 and 0.7071). Under their native protocols, the baselines score prompt-and-seed queries, whereas PatchRecover measures recovery of the candidate's content. These experimental results therefore highlight the value of direct image evidence.

Table 2: Expanded image detection (AUC and TPR@1%FPR). Both PatchRecover variants report mean ± SD over four regeneration seeds. Best and second-best are in bold and underlined, respectively.

	SD1		SD2	
Method	AUC ↑	TPR ↑	AUC ↑	TPR ↑
Ren et al.	0.9594	0.2682	0.9897	0.6435
Wen et al.	<u>0.9954</u>	<u>0.9461</u>	<u>0.9957</u>	**0.9954**
Jeon et al.	0.9883	0.8439	0.9851	0.7326
Chen et al.	0.9937	0.9305	0.9952	0.9861
Asthana and Belagiannis	0.9842	0.8294	0.9880	0.9109
Kim and Lee	0.9948	0.9004	0.9934	0.9618
PatchRecover	**0.9980**±0.0002	**0.9802**±0.0027	**0.9968**±0.0004	<u>0.9928</u>±0.0024
PatchRecover-PF	0.9712±0.0024	0.7628±0.0243	0.9914±0.0020	0.9462±0.0250

4.3 Image-level memorization detection

Because the mismatch test uses a small set of SD1 prompts, we next ask whether PatchRecover remains effective on the full benchmark and on SD2, keeping group-B images among the negatives. For groups A, B, and C, SD1 contains 482, 18, and 500 prompts, and SD2 contains 216, 3, and 219. With four images per prompt, this gives 4,000 SD1 and 1,752 SD2 images, with group-A images as positives and group-B and group-C images as negatives. Since group B is a small fraction of the negative class, this evaluation complements the focused mismatch test.

PatchRecover achieves the highest mean AUC on both SD1 and SD2 and the highest mean TPR on SD1 (Table 2). Unlike the more challenging mismatch setting (Table 1), where PatchRecover outperforms the baselines by a large margin, most methods here perform competitively, with most baselines already exceeding 0.98 AUC. Wen et al. have slightly higher SD2 TPR (0.9954 versus 0.9928). PatchRecover-PF trades some low-FPR sensitivity, mainly on SD1, for prompt-free detection. PatchRecover remains effective at benchmark scale. Prompt conditioning can facilitate more precise recovery of memorized content, but the prompt-free results show that discrimination does not rely solely on the prompt. The image itself also provides a strong recovery signal.

4.4 Patch-level memorization detection

Since image-level detection in PatchRecover relies on local evidence, we check whether individual patch scores can already distinguish memorization. The evaluation uses the same 20 SD1 prompt draws and comparisons as Section 4.2. Because patch-level ground truth is unavailable, each patch inherits its image's label. These labels are a noisy proxy because a memorized image need not be memorized in every region. For A_{18} versus B, each draw contains 2,304 patches per class, and for A_{36} versus $B + C$, 4,608. Appendix E details configuration selection and evaluation.

For A_{18} versus B, PatchRecover leads both AUC and TPR. For A_{36} versus $B + C$, it leads TPR, and its AUC ranks second behind Kim and Lee (Table 3). Individual recovery margins thus already carry detection signal, although their TPR is lower than that of the image scores. This comparison uses separately selected patch and image configurations, so Section 4.6 examines aggregation with the edit and distance held fixed.

4.5 Qualitative recovery examples

To show what the recovery margin captures, we inspect whether an edited patch returns toward the no-edit reference or retains the edit. In Figure 4, memorized regions in Regen. approach Ref., whereas non-memorized regions stay closer to Edit. This local contrast is what PatchRecover measures and what Section 4.4 observes quantitatively. Appendix F shows more examples across all five edits and both models.

Table 3: Patch-level detection with image-inherited labels (AUC and TPR@1%FPR). Mean ± SD over 20 A-prompt draws, with both PatchRecover variants first averaged over four regeneration seeds per draw. Best and second-best are in bold and underlined, respectively.

	A_{18} vs. B		A_{36} vs. $B + C$	
Method	AUC ↑	TPR ↑	AUC ↑	TPR ↑
Ren et al.	0.7914±0.0390	0.2669±0.0630	0.8924±0.0190	0.3763±0.0546
Wen et al.	0.8189±0.0370	0.3398±0.0810	0.8771±0.0239	<u>0.4004</u>±0.0769
Jeon et al.	0.8292±0.0144	0.0589±0.0169	0.9046±0.0075	0.0897±0.0179
Chen et al.	0.8137±0.0388	<u>0.3468</u>±0.0891	0.8654±0.0258	0.3911±0.0794
Asthana and Belagiannis	0.8246±0.0235	0.2030±0.0267	0.8792±0.0128	0.2620±0.0326
Kim and Lee	<u>0.8649</u>±0.0245	0.1642±0.0398	**0.9292**±0.0146	0.2239±0.0472
PatchRecover	**0.8792**±0.0143	**0.3642**±0.0366	<u>0.9093</u>±0.0091	**0.4307**±0.0409
PatchRecover-PF	0.7415±0.0264	0.1794±0.0254	0.8077±0.0160	0.2160±0.0376

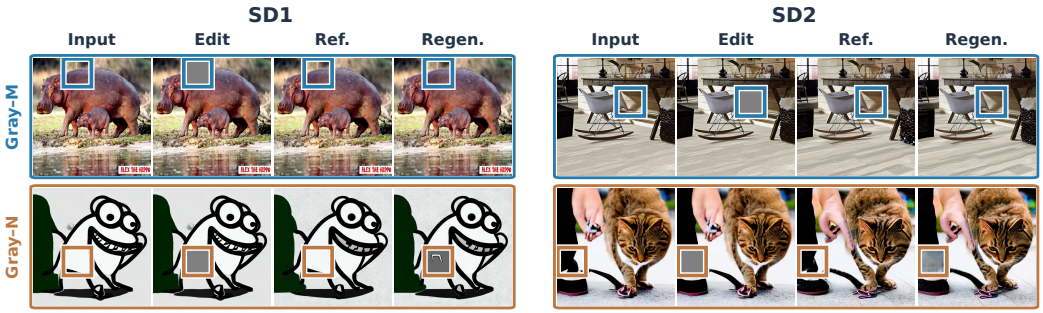

Figure 4: Selected gray-edit recovery examples on SD1 (left) and SD2 (right). Each four-image block shows Input, Edit, Ref., and Regen. Blue denotes M and orange denotes N.

4.6 Ablations and efficiency

We ask which score-construction choices (edits, distances, references, and aggregation) and regeneration choices (seeds, steps, strength, conditioning, and region sampling) affect performance and cost (Appendix D). PatchRecover is stable across the four regeneration seeds, with SDs of at most 0.0004 (AUC) and 0.0027 (TPR) on the expanded benchmarks (Table 2). Aggregation, in contrast, has a clear effect. For example, on SD1, top-k aggregation with k from 1 to 3 outperforms summing all regions, as PatchRecover avoids diluting the strongest regional evidence with weaker responses.

In terms of efficiency, the full configuration costs 17 regenerations per image (1 reference and 16 edited inputs). On the balanced ablation cohorts, querying 8 regions cuts regeneration count by 47.1%, from 17 to 9, with mean AUCs of 0.9923 on SD1 and 0.9962 on SD2 (Table 12). Compared with the full-region AUCs of 0.9990 and 0.9965 in the same experiment, this modest decrease offers a practical trade-off between regeneration cost and detection performance.

5 Conclusion

We introduced *PatchRecover*, an image-centric memorization detection method based on local counterfactual patch recovery. It requires no original prompt, seed, or denoising trajectory, making it practical when only generated images are available. We interpret this recovery through a local recovery-region analysis, showing that more frequent region entry and consistently stronger recovery separate memorized from non-memorized patches. PatchRecover achieves mean image-level AUCs of 0.9980 on SD1 and 0.9968 on SD2. The mismatch experiments confirm the value of direct image evidence. Prompt-free detection remains effective despite reduced low-FPR sensitivity. Appendix G discusses limitations and future directions.

AI USE STATEMENT

We used generative AI tools for tasks with recommended disclosure under ICLR policy: summarizing or analyzing existing literature, drafting parts of the manuscript, editing the manuscript to improve readability, and identifying relevant literature. We have not used generative AI tools for any task requiring disclosure. We have reviewed all AI-assisted contributions for accuracy. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Diffusion memorization raises privacy and copyright concerns. We present aggregate detection results and selected generated examples to document recovery behavior. These examples do not establish ownership or verified training-data provenance. The method is intended for studying memorization, but could also inform detector evasion. Any release of images or associated metadata should be reviewed for privacy and licensing risks.

REPRODUCIBILITY STATEMENT

Section 4 describes the experimental setup and evaluation protocols. Appendix A details benchmark construction, Appendix C records checkpoints and implementation choices, and we will release code upon acceptance. Appendix B provides additional theoretical and empirical analysis.

REFERENCES

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. *arXiv preprint arXiv:2303.08774*, 2023.

Rohan Asthana and Vasileios Belagiannis. Detecting and mitigating memorization in diffusion models through anisotropy of the log-probability. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HTPGy5ydAY.

Nicolas 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 *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, Anaheim, CA, August 2023. USENIX Association. ISBN 978-1-939133-37-3. 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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=p4cLtzk4oe.

CompVis. Stable Diffusion v1.4 model card, 2022. URL https://huggingface.co/CompVis/stable-diffusion-v1-4.

Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Yuanmin Huang, Mi Zhang, Chen Chen, Feifei Li, Geng Hong, Xiaoyu You, and Min Yang. Broken memories: Detecting and mitigating memorization in diffusion models with degraded generations. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, pp. 1932–1943, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3817770. URL https://doi.org/10.1145/3770855.3817770.

Dongjae Jeon, Dueun Kim, and Albert No. Understanding and mitigating memorization in generative models via sharpness of probability landscapes. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EW2JR5aVLm.

Yue Jiang, Haokun Lin, Yang Bai, Bo Peng, Zhili Liu, Yueming Lyu, Yong Yang, Xingzheng, and Jing Dong. Image-level memorization detection via inversion-based inference perturbation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vwOq7twk7L.

Gwangho Kim and Sungyoon Lee. Localizing memorized regions in diffusion models via coordinate-wise curvature differences. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=x16DrZepIl.

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.

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

Zhe Ma, Qingming Li, Xuhong Zhang, Tianyu Du, Ruixiao Lin, Zonghui Wang, Shouling Ji, and Wenzhi Chen. An inversion-based measure of memorization for diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16959–16969, 2025. doi: 10.1109/ICCV51701.2025.01575.

Manojb. Stable Diffusion 2 base model card, 2022. URL https://huggingface.co/Manojb/stable-diffusion-2-base.

Tomoya Matsumoto, Takayuki Miura, and Naoto Yanai. Membership inference attacks against diffusion models. In *2023 IEEE Security and Privacy Workshops (SPW)*, pp. 77–83, 2023. doi: 10.1109/SPW59333.2023.00013.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A Self-Supervised Descriptor for Image Copy Detection . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14512–14522, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01413. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01413.

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 *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXVII*, pp. 340–356, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72979-9. doi: 10.1007/978-3-031-72980-5_20. URL https://doi.org/10.1007/978-3-031-72980-5_20.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042.

Xinyue Shen, Yiting Qu, Michael Backes, and Yang Zhang. Prompt stealing attacks against Text-to-Image generation models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 5823–5840, Philadelphia, PA, August 2024. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/shen-xinyue.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ICML'15, pp. 2256–2265. JMLR.org, 2015.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion Art or Digital Forgery? Investigating Data Replication in Diffusion Models . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6048–6058, Los Alamitos, CA, USA, June 2023a. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.00586. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.00586.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=HtMXRGbUMt.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Linoy Tsaban and Sayak Paul. Tuxemon, 2024. URL https://huggingface.co/datasets/diffusers/tuxemon.

Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pp. 1398–1402 Vol.2, 2003. doi: 10.1109/ACSSC.2003.1292216.

Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, April 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861. URL https://doi.org/10.1109/TIP.2003.819861.

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.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068.

A Image-Level Benchmark Construction

A.1 Prompt and generation selection

Candidate memorized prompts come from the released resources of Webster (2023). Because the corresponding source images were not consistently retrievable, we use generated-image consistency as the selection criterion. Each prompt produces 128 candidate images under a fixed sampling protocol. An edge connects two generations when their SSCD cosine similarity is strictly greater than 0.7. We retain the first four-image clique for which all six pairs pass. The filter retains 482 of 500 SD1 prompts and 216 of 219 SD2 prompts. It favors conspicuous cross-seed repetition and may omit weaker or isolated cases.

A.2 Trajectory labels and their interpretation

Figure 2 uses all 16 grid patches from each of four selected images per prompt in the expanded cohorts. SD1 contains 482 M prompts and 518 N prompts, giving 30,848 M and 33,152 N trajectories. SD2 contains 216 M prompts and 222 N prompts, giving 13,824 M and 14,208 N trajectories. The N groups include all 500/219 non-memorization prompts and 18/3 memorization-associated prompts with prompt–image mismatch, respectively. Their selected images receive non-memorized labels because they fail the cross-seed repetition filter described above. All patches inherit their image's label. Class means give equal weight to every recorded trajectory, and entry probabilities use individual trajectories, allowing unequal class sizes.

Labeling limitations. Cross-seed filtering helps refine the provided memorization labels by identifying samples whose regeneration behavior is inconsistent with their original labels. However, direct verification against the training data is often not possible, as much of the original training data is unavailable. The filter may also miss local memorization that does not produce consistent cross-seed repetition. Such uncertain negatives account for $18/518 = 3.5\%$ of SD1 negatives and $3/222 = 1.4\%$ of SD2 negatives. These fractions indicate uncertainty under our labeling protocol rather than measured labeling-error rates. At the patch level, labels are inherited from their corresponding images and have no independent region-level verification.

B Additional Recovery Analysis

B.1 Proof of Proposition 1

Proof. Condition on the entry sampler state $\boldsymbol{s}_{i,t_0}$ and define $e_t = \mathbb{E}[d(\boldsymbol{p}_{i,t}, \boldsymbol{p}_i) \mid \boldsymbol{s}_{i,t_0}]$. By construction, $\boldsymbol{s}_{i,t}$ contains the full state required for the next reverse update. Hence, the reverse process is Markov with respect to $\boldsymbol{s}_{i,t}$. By the Markov property and the tower property,

$$e_{t-1} = \mathbb{E}[\mathbb{E}[d(\boldsymbol{p}_{i,t-1}, \boldsymbol{p}_i) \mid \boldsymbol{s}_{i,t}] | \boldsymbol{s}_{i,t_0}] .$$

Applying Eq. 4 and $\zeta_t \le (1-\lambda_t)\epsilon_i$ gives

$$\begin{aligned} e_{t-1} &\le \lambda_t e_t + \zeta_t \\ &\le \lambda_t e_t + (1-\lambda_t)\epsilon_i = \epsilon_i + \lambda_t(e_t - \epsilon_i). \end{aligned} \tag{7}$$

Equivalently,

$$e_{t-1} - \epsilon_i \le \lambda_t (e_t - \epsilon_i).$$

Recursively applying this inequality from t_0 to 0 yields

$$e_0 - \epsilon_i \le \left(\prod_{t=1}^{t_0} \lambda_t \right) (e_{t_0} - \epsilon_i).$$

At the entry state, $e_{t_0} = d(\boldsymbol{p}_{i,t_0}, \boldsymbol{p}_i)$, which gives Eq. 5. Since $\prod_{t=1}^{t_0} \lambda_t < 1$, whenever $d(\boldsymbol{p}_{i,t_0}, \boldsymbol{p}_i) > \epsilon_i$, the excess above ϵ_i is strictly reduced, establishing the stated recovery bound. □

B.2 Recovery gaps and mean separation

Table 4 separates the final recovery gaps by class and entry status, using the 64,000 SD1 and 28,032 SD2 trajectories in Figure 2. Each cell averages individual trajectories under the same Gray/Latent L2 configuration and fitted threshold used in that figure. M patches have smaller gaps in both entry groups. Entry is also associated with a smaller gap within each class.

Table 4: Mean recovery gaps by entry status and class. Smaller values indicate stronger recovery.

	SD1		SD2	
Entry status	M	N	M	N
Entry ($E = 1$)	-0.4047	0.3936	-0.3713	0.4161
Non-entry ($E = 0$)	0.4021	0.7741	0.5106	0.7384

Implication for mean separation. The three orderings in Section 3.3 are sufficient to compare the class means without assuming equal conditional gaps. To see this, define the intermediate average $q = \alpha_0^T \mu_{1,1}^T + (1 - \alpha_0^T)\mu_{0,1}^T$, which weights the M conditional means by the N entry frequency. Because M gaps are smaller in both groups, $\mathbb{E}[D_T \mid Y = 0] > q$. Because entry is more frequent for M and its entering gaps are smaller, $q > \mathbb{E}[D_T \mid Y = 1]$. Combining these inequalities gives

$$\mathbb{E}[D_T \mid Y = 1] < \alpha_0^T \mu_{1,1}^T + (1 - \alpha_0^T)\mu_{0,1}^T < \mathbb{E}[D_T \mid Y = 0],$$

and hence $\Delta_D^T < 0$ under the stated conditions. The recorded trajectory averages satisfy these conditions.

B.3 Sensitivity to the recovery threshold

We evaluate 101 equally spaced thresholds from $0.5\rho^\star$ to $1.5\rho^\star$, where $\rho^\star$ is the previously fitted threshold, without refitting or changing trajectories. M entry probability exceeds N entry probability at every tested threshold on both models (Figure 5). The smallest entry-probability gaps over this range are 0.0810 on SD1 and 0.0755 on SD2.

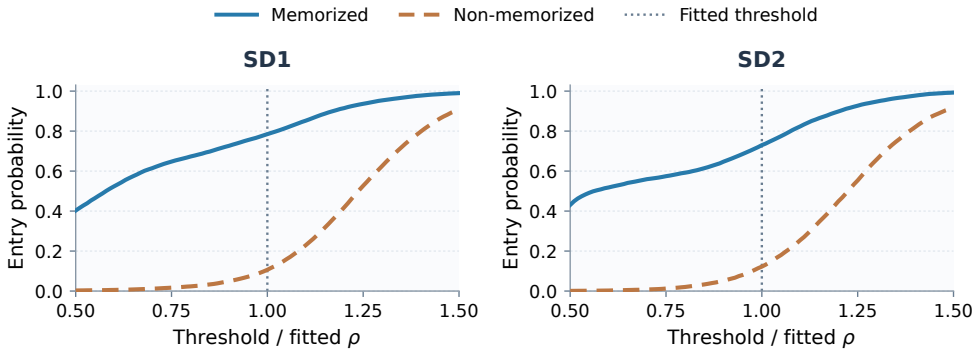

Figure 5: Entry probabilities across a threshold sweep. The dotted line marks the fitted threshold.

B.4 Late-stage regeneration behavior

Figure 6 illustrates late-stage regeneration drift using four selected rebound examples and one stable comparison from the SD1 memorized-image set. These examples are selected from a screen of 64 images (32 memorized and 32 non-memorized). They illustrate contrasting trajectories rather than estimate how often rebound occurs.

We regenerate each image without a local edit, using its source prompt and seed, the native PNDM scheduler (Liu et al., 2022), 50 sampling steps, strength 0.5, and guidance 7.5. The columns show

directly decoded intermediate latents. Their indices count remaining denoising updates, with 0 denoting the final output. To identify rebound, we compare the final full-image L2 distance to the original with the smallest distance among updates 1–5. Here L2 is the root-mean-square pixel difference on $[0, 1]$. Blue borders indicate this earlier minimum, while orange borders indicate the final output for rebound examples. In contrast, the last row remains stable through the final update. These examples show that regeneration can drift away from the original image during the final denoising steps, motivating the nonzero recovery floor in our local recovery model. This experiment characterizes general regeneration dynamics rather than the patch-level recovery effect itself.

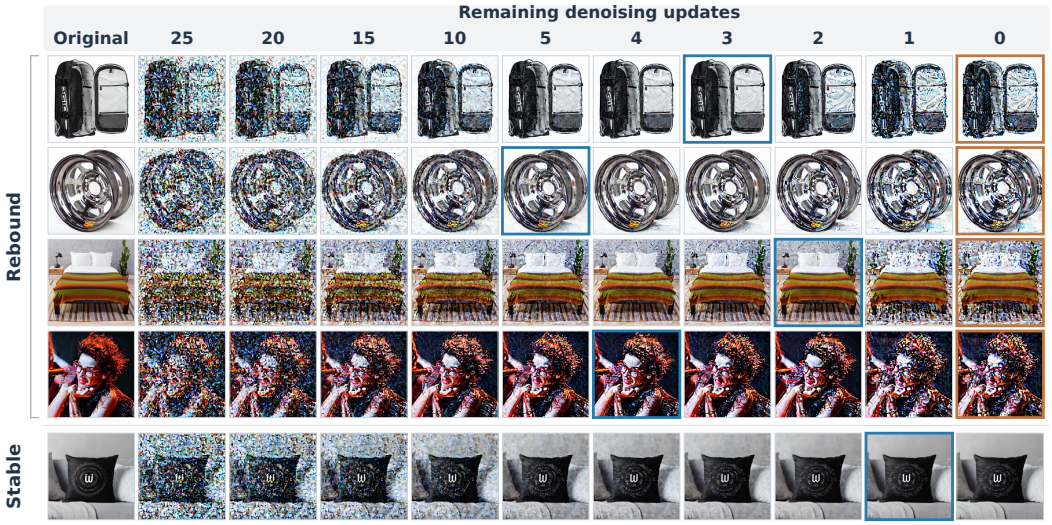

Figure 6: Unedited regeneration trajectories with four rebound examples (top) and one stable example (bottom). Columns count remaining denoising updates, with 0 denoting the final output.

B.5 Trajectory-plot conventions

- **Thresholds (Figure 2(a)).** Fitted recovery-region thresholds are shared across denoising steps. Dashed guides mark crossings by the class means, not individual entry probabilities.
- **Boxplots (Figure 2(b)).** Boxplots show medians and interquartile ranges with 1.5-IQR whiskers and a displayed subsample of trajectories.
- **Bootstrap intervals (Figure 2(c)).** Whole prompts are resampled while the fitted thresholds remain fixed.

C Implementation and Evaluation Protocols

C.1 PatchRecover implementation

Regeneration and region queries. Candidate images are fixed 512×512 PNGs. We retain each checkpoint's native scheduler: PNDM for SD1 and DDIM for SD2. Regeneration uses 50 inference steps, strength 0.5, and classifier-free guidance 7.5, with half-precision GPU inference. Each image has 16 non-overlapping grid regions. A no-edit regeneration is shared across all region queries, giving 17 regenerations per image. Each edited branch uses the same randomness and sampling settings as this reference. PatchRecover-PF uses empty text conditioning. The formulation permits other region sizes and masks, including non-square supports, but the present experiments evaluate uniform grids only.

Interventions and distances. Only the queried region is edited, leaving its surrounding context intact. Gray replaces the entire region with RGB value $(128, 128, 128)$. Neighbor copies a horizontally adjacent region, while Flip mirrors the region horizontally. JPEG applies compression at quality 20,

Algorithm 1 PatchRecover Inference

Require: Image $\boldsymbol{x}$, condition c, intervention T, dissimilarity d, randomness ξ, aggregation size k
Ensure: Image score $S_k(\boldsymbol{x})$ and patch scores $\{M_T(\boldsymbol{p}_i)\}_{i=1}^n$
$\hat{\boldsymbol{x}}^0 \leftarrow R_\theta(\boldsymbol{x}, c, \xi)$
for $i = 1, \ldots, n$ **do**
  $\tilde{\boldsymbol{p}}_i \leftarrow T(\boldsymbol{p}_i)$
  Replace region i in $\boldsymbol{x}$ to obtain $\boldsymbol{x}_i$
  $\hat{\boldsymbol{x}}_i \leftarrow R_\theta(\boldsymbol{x}_i, c, \xi)$
  Extract $\hat{\boldsymbol{p}}_i = \hat{\boldsymbol{x}}_i[i]$ and $\hat{\boldsymbol{p}}_i^0 = \hat{\boldsymbol{x}}^0[i]$
  Compute $M_T(\boldsymbol{p}_i)$ using Equation 1
end for
Compute $S_k(\boldsymbol{x})$ from the top-k patch scores
return $S_k(\boldsymbol{x})$ and $\{M_T(\boldsymbol{p}_i)\}_{i=1}^n$

Table 5: Baseline configurations with native signals and query schedules.

Method	Selected configuration and image-level signal
Ren et al. (2024)	**MemAttn-D.** Mean late-step token entropy plus the change in padding-token entropy from the initial step.
Wen et al. (2024)	**All-step score.** Whole-image L2 norm of the conditional-minus-unconditional noise prediction, averaged over all 50 DDIM steps.
Jeon et al. (2025)	**Sharpness.** Whole-image L2 norm of the conditional-minus-unconditional Hessian response at high noise.
Chen et al. (2025)	**Bright Ending.** Attention-weighted prediction-difference norms, normalized by mean attention and averaged over steps.
Asthana and Belagiannis (2026)	**Anisotropy.** Weighted combination of the high-noise guidance norm and spatially averaged low-noise alignment.
Kim and Lee (2026)	**Covariance curvature.** Sum over the coordinate-wise curvature map gives one image score.

and Blur uses radius 8. Scoring uses the same region as the intervention. We evaluate L2 and Latent L2, implemented as root-mean-square errors on aligned pixel and VAE-latent patches, respectively. Other distances are $1-$SSIM (Wang et al., 2004), $1-$MS-SSIM (Wang et al., 2003), LPIPS (Zhang et al., 2018), and one minus SSCD cosine similarity (Pizzi et al., 2022). Each configuration uses one distance throughout. Differently scaled distances are not combined. SSCD is included as a detection distance. Its use between paired recovery outputs is distinct from the cross-seed image comparisons used to construct the benchmark.

Configurations and aggregation. The expanded image configurations are Neighbor/L2 for SD1 and Gray/L2 for SD2. PatchRecover-PF uses Blur/Latent L2 and Gray/LPIPS, respectively. All use the recovery margin in Eq. 1 and top-2 aggregation. The implementation stores top-k sums rather than the averages in Eq. 2. For fixed k, this positive scaling leaves AUC and TPR unchanged. ACC thresholds are fitted on the same score scale.

C.2 Baseline implementations

Table 5 records the single paper-primary variant used for each baseline. All methods use the same target checkpoints: `CompVis/stable-diffusion-v1-4` and `Manojb/stable-diffusion-2-base`, a mirror of the SD2 base release (not SD2.1).

Adapters preserve the released score direction and do not select a sign using evaluation labels. Two methods required a small adaptation to yield a single image score: the Wen adapter follows all DDIM reverse steps and averages the conditional–unconditional prediction norm, and the Asthana adapter combines the published high-noise and low-noise terms. The remaining four baselines use their published scoring procedure unchanged. We use each method's native protocol rather than matching its query count to PatchRecover. The main table therefore compares detection quality, not

Table 6: Matched SD1 comparisons with one image per prompt. PatchRecover variants report mean ± SD across four regeneration seeds.

	A_{18} vs. B		A_{36} vs. $B + C$	
Method	AUC ↑	TPR@1%FPR ↑	AUC ↑	TPR@1%FPR ↑
InvMM	0.7531	0.1111	0.8519	0.1944
PatchRecover	**0.9946**±0.0046	**0.9306**±0.0699	**0.9944**±0.0030	**0.8681**±0.0829
PatchRecover-PF	0.9035±0.0096	0.5000±0.0454	0.9163±0.0068	0.3542±0.0616

equal-cost efficiency. Exact repository revisions and executable configurations will accompany the code release.

C.3 Small-scale comparison with InvMM

To compare against an optimization-based image detector within a tractable evaluation budget, we evaluate InvMM and both PatchRecover variants on the same 72 SD1 images. A full-dataset evaluation is computationally prohibitive because InvMM performs costly per-image inversion through iterative optimization. We therefore evaluate InvMM and both PatchRecover variants on the same subset under a tractable computational budget. The subset contains 36 group-A, 18 group-B, and 18 group-C images, using the first prespecified prompt draw from the mismatch evaluation and one saved image per prompt, selected by the lowest source seed. This gives 36 positive and 36 negative images. The nested A_{18} versus B comparison uses the first 18 group-A prompts in the same draw and all 18 group-B images.

InvMM uses one optimization run per image with up to 2,000 steps, 75 learned conditioning tokens, and 50-step DDIM verification. Every 50 optimization steps, reproduction is checked against an SSCD similarity threshold of 0.5 using eight generations at each of seven guidance scales. Successful reproduction requires all eight generations at one scale to pass the check. Up to $40 \times 56 = 2{,}240$ verification generations are therefore possible per candidate, in addition to optimization. Early stopping reduces this configured maximum, which is not a measured average. The implementation scores successful candidates by $1/(1 + \mathrm{KL})$, where KL measures divergence of the optimized noise distribution from its Gaussian prior, and assigns zero to unsuccessful candidates. Of the 72 images, 71 pass verification. The remaining group-C image stays in the evaluation with its zero score.

PatchRecover uses Neighbor/L2 and PatchRecover-PF uses Blur/Latent L2, both with paired no-edit regeneration and top-2 aggregation. These are the existing SD1 image-level configurations, without tuning on this subset. Saved scores are reused after verifying image hashes. Metrics are computed separately for four regeneration seeds and summarized by their mean and sample standard deviation. InvMM and PatchRecover-PF do not receive the original prompt. PatchRecover uses prompt conditioning. With 18 or 36 negatives, TPR at 1% FPR permits no false positives in either comparison.

Both PatchRecover variants attain higher AUC and TPR than InvMM in both comparisons (Table 6). PatchRecover-PF also exceeds InvMM without access to the original prompt, demonstrating discrimination from local recovery without per-image optimization on these matched images.

C.4 Metric definitions

Image ACC and B specificity use the frozen, prompt-disjoint thresholds described in Appendix C.5. A score at or above the threshold predicts memorization. AUC uses continuous scores. TPR@1%FPR is the maximum empirical TPR at FPR no greater than 1%. For our methods, expanded-benchmark metrics are summarized across regeneration seeds. Mismatch metrics are averaged over seeds within each draw, then summarized across draws as described in Appendix C.5. Configuration selection precedes the prompt-disjoint threshold fits.

Table 7: Image-level threshold transfer: ACC on both mismatch comparisons and specificity on B. Mean ± SD across 20 prompt-disjoint threshold fits. Both PatchRecover variants first average metrics over four regeneration seeds per draw.

Method	A_{18} vs. B ACC ↑	A_{36} vs. $B + C$ ACC ↑	B specificity ↑
Ren et al.	0.6642±0.0065	0.8168±0.0042	0.3333±0.0000
Wen et al.	0.7667±0.0102	0.8649±0.0079	0.5417±0.0000
Jeon et al.	0.7552±0.0159	0.8585±0.0091	0.5556±0.0000
Chen et al.	0.6889±0.0105	0.8111±0.0081	0.3938±0.0130
Asthana and Belagiannis	0.8007±0.0216	0.8722±0.0140	0.6861±0.0131
Kim and Lee	0.6698±0.0094	0.8174±0.0076	0.3472±0.0000
PatchRecover	**0.8476**±0.0101	**0.9124**±0.0070	0.7071±0.0081
PatchRecover-PF	0.7990±0.0282	0.8487±0.0229	**0.7097**±0.0028

C.5 Prompt-image mismatch evaluation

Cohorts and image thresholds. For each of 20 draws, A_{18} contains the first 18 prompts in the draw defining A_{36}. Draws are shared across all methods and are not selected according to their results. The two comparisons therefore share a nested positive cohort and the same group B. For each draw, we fit image-level thresholds after excluding every A_{36}, B, and C prompt from both classes of the full SD1 benchmark, matching Unicode-NFKC-normalized, case-folded text with collapsed whitespace. The remaining 446 positive and 482 negative prompts provide 3,712 fitting images. Both comparisons use this same prompt-disjoint set, excluding all four images of each held-out prompt. Maximizing fitting-set accuracy gives one frozen threshold per draw, method, configuration, and regeneration setting. PatchRecover uses Neighbor/L2 and PatchRecover-PF uses Blur/Latent L2, both with top-2 aggregation. These configurations supply all three metrics. PatchRecover-PF's threshold is fitted on Latent L2 scores, not transferred from LPIPS. Threshold fitting has no test-prompt overlap, but prior configuration selection is not independent of these data. With only 72 negative images in A_{18} versus B, empirical TPR@1%FPR allows no false positives.

Image-level threshold transfer. Table 7 complements the ROC results with frozen-threshold ACC and specificity on the fixed non-memorized group B. Thresholds are fitted after excluding every test prompt. Both comparisons share the fitting set. B specificity varies across threshold fits, not across B resampling, and should be read alongside two-class ACC rather than as a standalone detector score. PatchRecover attains the highest mean ACC in both comparisons (0.8476 and 0.9124), while PatchRecover-PF has the highest mean B specificity (0.7097 versus 0.7071 for PatchRecover). Removing prompt conditioning yields slightly fewer false alarms on B at the fitted thresholds, but lower two-class ACC.

Averaging across draws and seeds. For each method, Table 1 reports the mean and sample standard deviation across the 20 prompt draws. For both PatchRecover variants, metrics from the four regeneration seeds are averaged within each draw before this summary. Scores are not averaged to form an ensemble. Averaging in this order separates variation across positive prompt cohorts from variation across regeneration seeds. B and C remain fixed, so the deviations do not represent uncertainty from sampling those groups. The four seeds are not treated as independent prompt draws. Within-draw regeneration variability can be summarized separately as the square root of the mean within-draw sample variance. The accompanying result files provide this statistic for ACC, AUC, and TPR.

D Image-Level Ablations and Efficiency

The full sweep covers five edits, six distances, three score formulas, and four aggregations, reported in Tables 8, 9, and 10, all under the original-seed regeneration setting. For each candidate image, the same regeneration seed is reused across compared settings, and the edited and no-edit branches share identical randomness, holding regeneration randomness fixed while varying the factor under study. It does not mean that every candidate image uses one common seed.

Table 8: Edit–distance sweep: image AUC with the recovery margin and top-2 aggregation. MS denotes MS-SSIM.

	SD1						SD2					
Edit	L2	Latent L2	1− SSIM	1− MS	LPIPS	1− SSCD	L2	Latent L2	1− SSIM	1− MS	LPIPS	1− SSCD
Gray	0.9963	0.9960	0.9955	0.9954	0.9962	**0.9961**	**0.9965**	**0.9947**	**0.9946**	**0.9947**	0.9942	**0.9956**
Neighbor	**0.9990**	0.9931	0.9812	0.9871	0.9924	0.9833	0.9957	0.9900	0.9865	0.9881	**0.9944**	0.9797
Flip	0.9981	0.9925	0.9756	0.9869	0.9858	0.6146	0.9964	0.9860	0.9847	0.9895	0.9934	0.6869
JPEG	0.3056	0.5093	0.3448	0.3059	0.5977	0.4247	0.1154	0.2430	0.1218	0.0587	0.4975	0.2905
Blur	0.9879	**0.9967**	**0.9970**	**0.9962**	**0.9973**	0.9841	0.9881	0.9924	0.9918	0.9891	0.9941	0.9928

D.1 Edit and distance sweep

Table 8 reports the complete recovery-margin and top-2 sweep. Each entry is image-level AUC under the original-seed protocol. L2, Latent L2, SSIM, MS-SSIM, LPIPS, and SSCD distances are evaluated independently. The strong gray, neighbor, flip, and blur results show that PatchRecover is not tied to one edit, whereas JPEG compression reverses the score ordering for several distances in this setting. The JPEG and Flip explanations below are hypotheses based on observed scores and method properties. Neither mechanism has been tested directly.

JPEG failure case. JPEG reverses the score ordering. With L2 the AUCs are 0.3056 on SD1 and 0.1154 on SD2. Recomputing the saved top-2 margins gives mean scores of 0.1783 (M) versus 0.2391 (N) on SD1, and 0.1497 versus 0.2404 on SD2. On the selected top-2 patches, both distance components are smaller for M, but the decrease in d_{edit} exceeds that in d_{ref}, reversing the margin ordering. Inspection of selected examples (Figure 7, JPEG rows) shows that JPEG preserves the underlying content while adding compression artifacts. Unlike a content-changing edit, it may therefore measure reconstruction changes rather than return from a distinct alternative. JPEG remains a failure case with its original score direction, without a label-informed sign reversal.

Flip and SSCD. Flip with SSCD is substantially weaker than Flip with L2 (AUC 0.6146/0.6869 versus 0.9981/0.9964 on SD1/SD2). SSCD uses horizontal flips in its contrastive training augmentations (Pizzi et al., 2022), encouraging invariance to this transformation. Such invariance can reduce the distinction between a patch and its flipped alternative, plausibly weakening the recovery margin.

D.2 Reference, score, and aggregation design

The primary recovery margin is M_T from Eq. 1, with $d_{\text{edit}} = d(\hat{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}_i)$ and $d_{\text{ref}} = d(\hat{\boldsymbol{p}}_i, \hat{\boldsymbol{p}}_i^0)$. We also evaluate d_{edit} alone and $-d_{\text{ref}}$ alone as diagnostic score families, but do not combine configurations across distance scales. The reference ablation in Table 9 compares the matched no-edit regeneration with direct reference to the source image while holding the primary edit, distance, margin, and top-2 aggregation fixed. Their performance is nearly identical. We retain the no-edit reference because it controls the variation caused by regeneration and matches the practical inference rule in Section 3.1.

The aggregation rows use the matched reference and the same primary configuration. On SD1, summing every region reduces AUC from 0.9990 to 0.9592, indicating that including weaker regional scores dilutes the signal. On SD2, AUC is less sensitive, although the all-region sum lowers TPR@1%FPR.

D.3 Regeneration hyperparameters and spatial granularity

Table 10 varies the inference-step count and image-to-image strength around the primary configuration. Ten steps reduce low-FPR sensitivity, while 50 and 100 steps are similar. Very weak intervention strength provides insufficient opportunity for counterfactual recovery, whereas strength 0.9 can overwrite local evidence. Strength 0.5 is therefore used throughout the main evaluation. The

Table 9: Reference and aggregation ablations (AUC/TPR), using the original generation seed.

Factor	Setting	SD1	SD2
Reference	no-edit regeneration	0.9990/0.9896	0.9965/**0.9931**
	source image	**0.9993/0.9917**	**0.9968/0.9931**
Aggregation	top-1 sum	0.9989/0.9881	0.9965/**0.9931**
	top-2 sum	**0.9990/0.9896**	0.9965/**0.9931**
	top-3 sum	0.9987/0.9865	0.9966/**0.9931**
	all-region sum	0.9592/0.8522	**0.9969**/0.9757

Table 10: Regeneration and spatial-granularity ablations (AUC/TPR).

Factor	Setting	SD1	SD2
Inference steps	10	0.9653/0.8517	0.9910/0.8727
	50	0.9990/0.9896	**0.9965/0.9931**
	100	**0.9994/0.9943**	**0.9965/0.9931**
Strength	0.1	0.7418/0.3013	0.7286/0.0012
	0.5	**0.9990/0.9896**	**0.9965/0.9931**
	0.9	0.9561/0.6432	0.8932/0.3137
Grid	2×2 best	0.9971/**0.9922**	0.9942/0.9722
	4×4 best	**0.9990**/0.9896	**0.9965/0.9931**

spatial-resolution rows report recovery-margin top-2 configurations at each resolution and show that the method is not restricted to one grid size.

D.4 Robustness to regeneration randomness

Table 11 compares the original generation seed with three source-independent regeneration seeds, labeled Seed-1, Seed-2, and Seed-3. "Original seed" denotes reuse of the seed that generated the candidate image, included as a reference condition rather than a detection requirement. All four regeneration seeds use the same images and score configuration. Paired branches always share randomness. The ablation uses the original balanced cohorts (482 prompts per class on SD1 and 216 per class on SD2), not the expanded cohorts in Table 2. SD1 PatchRecover-PF uses Blur/LPIPS in this ablation. PatchRecover is stable across these settings on both models. PatchRecover-PF also retains high AUC. Its SD2 low-FPR sensitivity varies more across seeds, indicating greater sensitivity to regeneration randomness without conditioning.

D.5 Random-region query budgets

To quantify the cost–accuracy tradeoff, let b denote the number of edited regions evaluated for each image. We sample a subset $\mathcal{S}_{b,r}(\boldsymbol{x})$ of the 16 benchmark regions and retain top-2 aggregation:

$$\widetilde{S}^{(r)}_{2,b}(\boldsymbol{x}) = \sum_{j=1}^{2} M_T(\boldsymbol{p}_{(j;\mathcal{S}_{b,r})}). \tag{8}$$

Here the ranking is restricted to the sampled regions, and the score uses the implementation sum described above. For each $b \in \{2, 4, 8\}$, we repeat evaluation over 50 reproducible random subsets sampled without replacement. The case $b = 16$ evaluates every region and is therefore deterministic. A budget of b regions requires $b + 1$ image-to-image queries: one shared no-edit reference and b edited regenerations. Relative to the full 17-query evaluation, the reduction is $1 - (b + 1)/17$. Table 12 reports the mean and sample standard deviation of the 50 trial-level metrics. Within each trial, a uniform random permutation of the 16 regions is drawn independently for each image. The first b regions form the queried subset, so budgets are nested within that trial. The standard deviation measures sensitivity to region selection, not uncertainty across images or prompts. The full-region result has no sampling variation.

Table 11: Image detection across four regeneration seeds.

	SD1				SD2			
	PatchRecover		PatchRecover-PF		PatchRecover		PatchRecover-PF	
Setting	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
Original seed	0.9990	0.9896	0.9851	0.8838	0.9965	0.9931	0.9928	0.9583
Seed-1	0.9987	**0.9907**	0.9859	**0.8968**	0.9965	0.9896	**0.9938**	**0.9826**
Seed-2	**0.9992**	0.9886	0.9858	0.8797	**0.9971**	0.9931	0.9899	0.9294
Seed-3	0.9987	0.9901	**0.9870**	0.8916	**0.9971**	**0.9954**	0.9901	0.9421

Table 12: Random-region detection: mean ± SD over 50 subset trials. The full-region result is deterministic.

			SD1		SD2	
b	Queries	Reduction	AUC	TPR@1%FPR	AUC	TPR@1%FPR
2	3	82.4%	0.9253±0.0032	0.7364±0.0107	0.9704±0.0039	0.8477±0.0156
4	5	70.6%	0.9713±0.0020	0.9027±0.0053	0.9924±0.0015	0.9674±0.0048
8	9	47.1%	0.9923±0.0011	0.9662±0.0040	0.9962±0.0005	0.9887±0.0016
16	17	0.0%	**0.9990**	**0.9896**	**0.9965**	**0.9931**

E Patch-Level Evaluation Protocol

Table 3 tests local recovery scores without image-level aggregation over 20 nested random A-prompt draws, with B/C held fixed. All patches in A inherit the memorized label, and all patches in B and C inherit the non-memorized label, yielding 2,304 and 4,608 patch instances per class, respectively. The evaluation measures separation under image-inherited labels. Assessing localization accuracy would require independent patch annotations.

Configuration selection. Both patch comparisons use the exact 20 A draws and fixed B/C images of Table 1. The exploratory sweep evaluates complete PatchRecover and PatchRecover-PF edit–distance configurations, each with all four regeneration seeds on every evaluated image. Configurations are ranked by patch AUC averaged first over seeds and then over all 20 draws, separately for each comparison. A single configuration supplies AUC and TPR in Table 3. No draws or regeneration settings are selected for favorable results. Both comparisons use Gray/L2 for PatchRecover and Gray/MS-SSIM for PatchRecover-PF (SSIM and MS-SSIM denoting their corresponding dissimilarities). Incomplete configurations are excluded. Configuration selection uses the evaluated cohorts, so the reported performance includes selection on these data.

Both PatchRecover variants first average seed-level metrics within each draw. All methods then report the across-draw mean and sample standard deviation using the procedure in Appendix C.5.

F Additional Qualitative Recovery Examples

Figure 7 extends Figure 4 to all five edits on both model families. For each edit, adjacent M and N rows contrast selected memorized and non-memorized examples. The examples retain the same query-region annotations used in Figure 4. Selection favors small paired regeneration differences for M and large differences for N, subject to visible-content and visual-diversity screening. Distinct prompts are used across the edit–model–category examples. Aggregate performance is reported separately in Table 8. JPEG can invert the detection score ordering, as discussed in Appendix D.

Each four-image block shows the original input, edited input, no-edit reference regeneration, and edited regeneration. The paired branches use the same randomness. Blue frames denote memorized examples and orange frames non-memorized examples. The smaller boxes mark the queried region. Gray replaces the whole queried patch with a constant gray value.

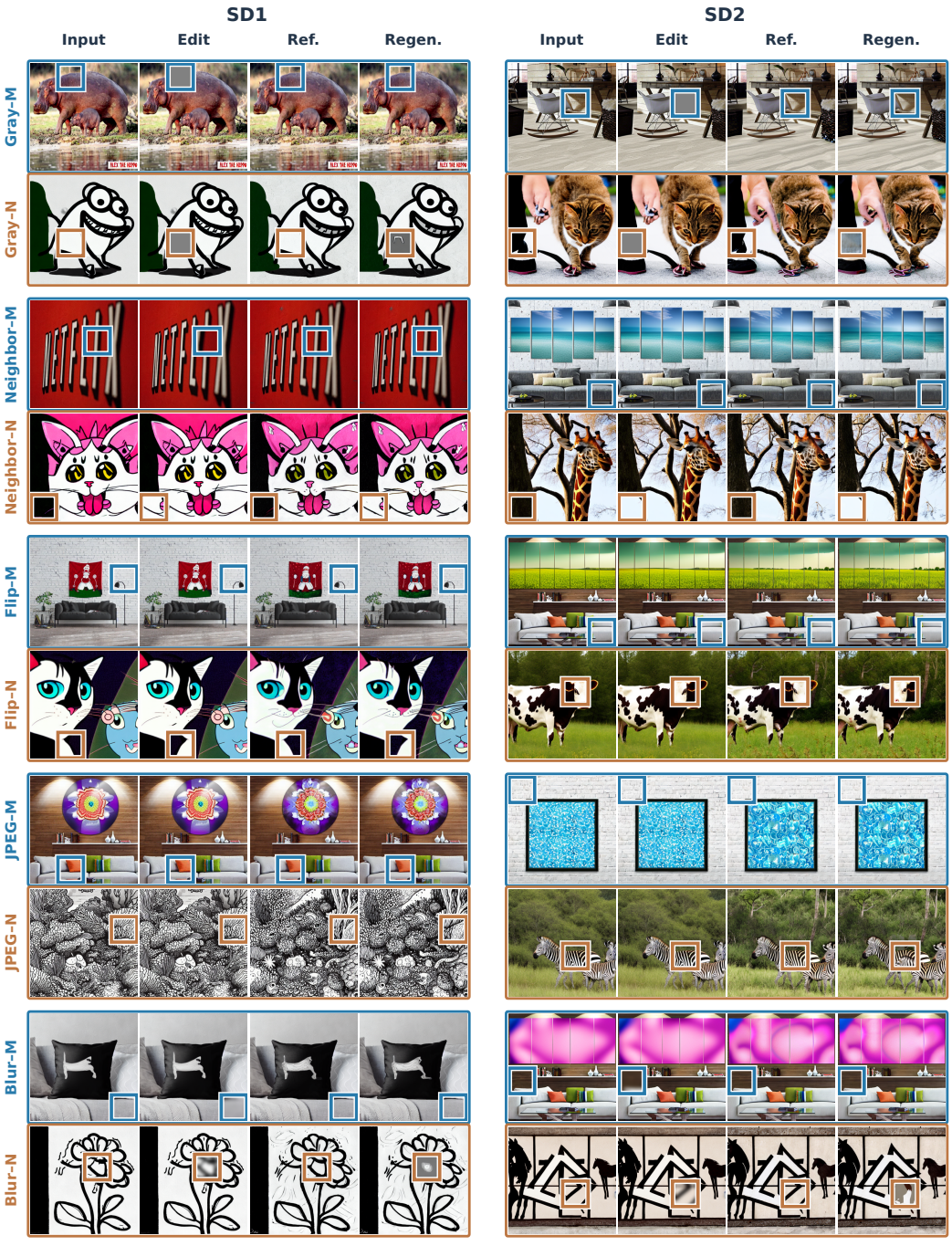

Figure 7: Selected recovery examples across five edits on SD1 (left) and SD2 (right). Each colored frame encloses one sample's four images: blue for M, orange for N.

G LIMITATIONS AND FUTURE DIRECTIONS

Our evaluation has several limitations. First, the evaluation covers two Stable Diffusion families. Broader datasets would enable evaluation on newer diffusion models. Second, cross-seed filtering helps refine the provided image-level labels, but direct verification against the original training data is often unavailable. Patch-level AUC and TPR use labels inherited from the source image and show strong discrimination under this protocol, but some patches from memorized images may not themselves be memorized due to the lack of region-level ground truth. Finally, baseline coverage is partly limited by the high computational cost of optimization-based detectors. Future work could expand model and baseline coverage and evaluate on datasets with verified training images and region-level memorization annotations.