MITIGATING SEQUENTIAL REAPPEARANCE IN DIFFUSION DATA-POINT UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion data-point unlearning is typically evaluated immediately after each deletion, even though subsequent requests may repeatedly update the same model. We identify sequential reappearance, a failure mode in which an instance that is initially judged to be forgotten later returns to the memorized regime without reuse of the deleted data or adversarial fine-tuning. To capture this behavior, we introduce a target-level evaluation protocol that tracks whether each target is forgotten immediately, remains forgotten at the end of the sequence, or reappears during subsequent deletions. We further find that targets that later reappear exhibit greater local recovery accessibility after deletion than targets that remain forgotten. We characterize this vulnerability as recovery accessibility and show that this post-deletion local geometry is associated with subsequent reappearance risk. Motivated by this finding, we propose a recovery-guided unlearning method that iteratively identifies the most recoverable region around each target and extends the deletion objective to that region. Experiments show that our method improves forgetting persistence across long deletion sequences while maintaining competitive generation quality, highlighting the importance of evaluating diffusion unlearning beyond immediate deletion efficacy.

1 INTRODUCTION

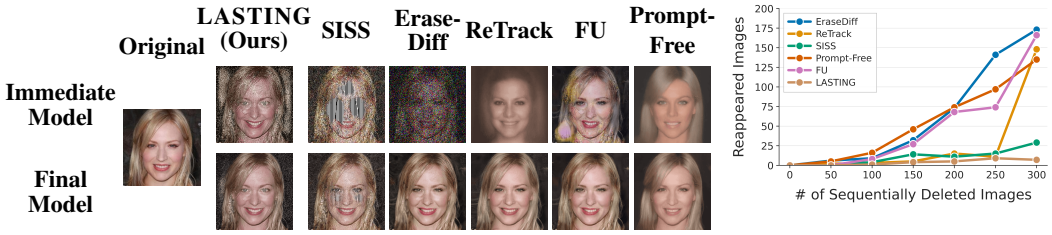
Diffusion models can memorize and reproduce individual training examples, raising privacy and copyright concerns when sensitive data must be removed (Somepalli et al., 2023a; Carlini et al., 2023). Diffusion data-point unlearning removes the influence of a specific training instance without retraining the model from scratch (Alberti et al., 2025; Shi et al., 2026). A reliable deletion should remain effective throughout subsequent updates to the same model. As shown in Figure 1a, however, a target that is forgotten at its immediate checkpoint can become memorized again after later deletion requests.

Existing evaluations predominantly assess each target immediately after its deletion (Alberti et al., 2025; Wu et al., 2025; Shi et al., 2026; Park & Park, 2026). Such stage-local evaluation overlooks the target's behavior under future model updates and can overestimate deletion reliability. We identify *sequential reappearance*, in which an initially forgotten target returns to the memorized regime during ordinary sequential deletion without reuse of the target, adversarial fine-tuning, or an explicit recovery attack.

To understand this failure, we examine the local denoising-loss geometry around each target immediately after deletion. We find that targets that later reappear have greater local recovery accessibility than those that remain forgotten, even when both satisfy the immediate forgetting criterion. We refer to this vulnerability as *recovery accessibility*. This finding shows that reliable unlearning should suppress not only the observed target but also nearby representations that preserve a route back to memorization.

Motivated by this observation, we propose *Low-loss Adaptive Search and Targeted Intervention in Neighborhood Geometry* (LASTING). LASTING repeatedly identifies the most recoverable nearby representation and uses it to guide the deletion update as the model changes. Across long sequential deletion trajectories, LASTING substantially improves forgetting persistence while maintaining competitive generation quality. These results show that successful diffusion unlearning requires forgetting to remain stable under subsequent deletion requests.

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.



(a) Qualitative comparison.

(b) Reappearance counts.

Figure 1: Sequential reappearance in CelebA-HQ. (a) Outputs generated immediately after unlearning each target and after processing all subsequent deletion requests. Existing methods can reproduce initially forgotten targets in the final model, whereas LASTING maintains forgetting after the full sequence of unlearning updates. (b) Number of reappeared images at each deletion checkpoint.

Contributions. Our main contributions are summarized as follows:

- We introduce a target-level protocol that evaluates both immediate and persistent forgetting under sequential diffusion unlearning.
- We identify *sequential reappearance* and show that it is associated with greater post-deletion recovery accessibility.
- We propose LASTING, which adaptively suppresses the most locally recoverable representation and substantially improves forgetting persistence while maintaining competitive generation quality.

2 Related Work

2.1 Memorization in Diffusion Models

Diffusion models can reproduce individual training examples, and extraction attacks have recovered memorized images at scale (Somepalli et al., 2023a; Carlini et al., 2023). Prior work studies how duplication, data coverage, and text conditioning affect replication and develops detection or inference-time mitigation methods (Somepalli et al., 2023b; Wen et al., 2024; Merger & Goldt, 2026; Han et al., 2025). Unlike these interventions, diffusion unlearning modifies the model to remove the influence of selected training data.

2.2 Diffusion Model Unlearning

Concept-level unlearning suppresses broad semantic content by editing model weights, aligning target representations with anchors, or selecting parameters associated with the target concept (Gandikota et al., 2023; Kumari et al., 2023; Zhang et al., 2024a; Fan et al., 2024; Heng & Soh, 2023; Lu et al., 2024; Li et al., 2025). Because deletion is defined semantically, these methods do not precisely remove an individual training example while preserving related content.

Data-point unlearning instead targets individual examples. EraseDiff redirects the reverse process, SISS balances forgetting and utility through an importance-sampled score objective, and ReTrack redirects a target trajectory toward retain-set neighbors (Wu et al., 2025; Alberti et al., 2025; Shi et al., 2026). Recent methods formulate unlearning through KL-constrained optimization or use edited surrogate targets without textual prompts (Khalafi et al., 2026; Lee et al., 2026b). These approaches focus primarily on forgetting immediately after deletion. Our work examines whether successfully deleted instances remain forgotten as the same model undergoes subsequent deletion updates.

2.3 Recovery and Sequential Unlearning

Recovery after unlearning has mainly been studied under explicit post-unlearning interventions, including fine-tuning on removed data or with unrelated prompts, malicious relearning, and adver-

sarial prompt attacks (George et al., 2025; Gao et al., 2025; Zhang et al., 2024c; Suriyakumar et al., 2026). Our setting concerns spontaneous reappearance during ordinary sequential deletion without additional fine-tuning, recovery attacks, or reuse of the removed instance.

Sequential diffusion unlearning has largely focused on semantic concepts and utility preservation across successive removals (Lee et al., 2026a; Thakral et al., 2025; George et al., 2026). Closest to our setting, prior work reports unlearning rebound in sequential style erasure and forgetting reversal in continual classifier unlearning (Zhang et al., 2024b; Park et al., 2026). Sequential data-point unlearning in diffusion models remains largely unexplored, despite its relevance to accumulating copyright and privacy requests.

2.4 Local Geometry in Machine Unlearning

Parameter-space sharpness. Loss-landscape sharpness is closely related to generalization and robustness (Keskar et al., 2017; Dinh et al., 2017). Sharpness-Aware Minimization and Adversarial Weight Perturbation measure loss variation under parameter perturbations, and recent unlearning methods use similar ideas to improve stability under relearning (Foret et al., 2021; Wu et al., 2020; Tang & Khanna, 2026; Malekmohammadi et al., 2025; Fan et al., 2025).

Data-space local geometry. Prior classifier studies examine residual knowledge around perturbed forget samples or reshape local decision boundaries with adversarial examples (Hsu et al., 2025; Ebrahimpour-Boroojeny et al., 2025). We instead study the data-space geometry of the diffusion denoising loss and identify nearby image representations that remain easy to denoise after deletion. We examine the association between this post-deletion recovery geometry and subsequent reappearance.

3 Analysis of Sequential Reappearance

3.1 Protocol and Metrics

Sequential unlearning setup. Let M_{θ_0} be a diffusion model and let $\mathcal{A} = (a_1, \ldots, a_K)$ be an ordered sequence of deletion targets. Given retain data $\mathcal{R}$, an unlearning algorithm $\mathcal{U}$ updates the model at stage i as

$$\theta_i = \mathcal{U}\left(\theta_{i-1}; a_i, \mathcal{R}\right), \qquad i = 1, \ldots, K. \tag{1}$$

For target a_i, θ_i is its *immediate model state*, whereas θ_K is the *final model state* after all subsequent requests. This distinction separates immediate forgetting from forgetting that persists throughout the deletion trajectory.

Persistence metrics. Let $s_i(\theta_j)$ denote the SSCD similarity (Pizzi et al., 2022) between target a_i and its recovery output from M_{θ_j}. We consider the target forgotten when $s_i(\theta_j) < \tau$ and use $\tau = 0.7$ following prior work (Somepalli et al., 2023b). We define Final Success and Final Reappearance as

$$\begin{aligned} \text{FinalSucc.@}\tau &= \frac{1}{K}\sum_{i=1}^{K} \mathbf{1}\left[s_i(\theta_K) < \tau\right], \\ \text{FinalReapp.@}\tau &= \frac{\sum_{i=1}^{K} \mathbf{1}\left[s_i(\theta_i) < \tau \wedge s_i(\theta_K) \geq \tau\right]}{\sum_{i=1}^{K} \mathbf{1}\left[s_i(\theta_i) < \tau\right]}. \end{aligned} \tag{2}$$

Final Success measures the fraction of targets forgotten under the final model. Final Reappearance measures the fraction of immediately forgotten targets that return to the memorized regime under the final model.

3.2 Reappearance under Sequential Deletion

We apply this protocol to $K = 300$ deletion requests using SISS, EraseDiff, ReTrack, Prompt-Free, and FU (Shi et al., 2026; Wu et al., 2025; Alberti et al., 2025; Khalafi et al., 2026; Lee et al., 2026b). Figure 1b shows that previously forgotten targets can return to the memorized regime as later deletions update the model, confirming that immediate forgetting does not ensure persistent forgetting.

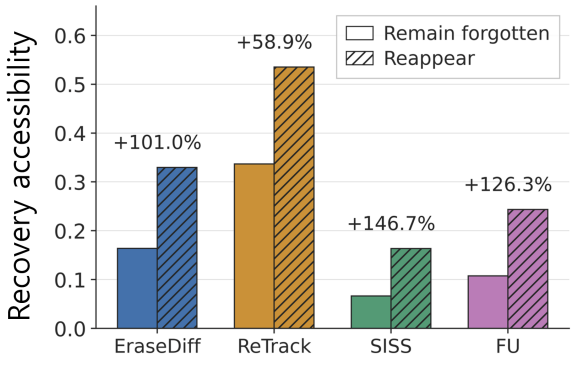


Figure 2: Recovery accessibility by outcome. Targets that subsequently reappear exhibit higher recovery accessibility after the initial unlearning step.

Table 1: Association between recovery accessibility and subsequent reappearance. ρ denotes Spearman correlation with maximum future SSCD, partial ρ controls for Immediate SSCD, and AUC denotes AUROC.

Method	ρ	Partial ρ	AUC
EraseDiff	0.528	0.529	0.740
ReTrack	0.436	0.381	0.750
SISS	0.452	0.234	0.707
FU	0.358	0.349	0.669

Recovery accessibility. To explain this behavior, we examine the local denoising-loss geometry around each target immediately after deletion. SSCD indicates whether the observed target is recoverable but does not reveal whether nearby representations remain easy to denoise. For each target a_i, we freeze the immediate model θ_i, construct a fixed bank $\mathcal{B}_i$ of timesteps and noise samples, and define the local denoising loss as

$$L_i(\delta) = \frac{1}{|\mathcal{B}_i|} \sum_{\xi \in \mathcal{B}_i} \ell_{\text{diff}}\left(\theta_i; a_i + \delta, \xi\right). \tag{3}$$

Within the bounded data-space neighborhood $\mathcal{D}_i = \{\delta : \|\delta\|_\infty \leq \epsilon, a_i + \delta \in [-1, 1]^d\}$, we approximate a low-loss neighbor and define *recovery accessibility* as

$$\widehat{\delta}_i \approx \arg\min_{\delta \in \mathcal{D}_i} L_i(\delta), \quad \text{RecAccess}_i = \max\left\{0, \frac{L_i(0) - L_i(\hat{\delta}_i)}{\max\{L_i(0), \eta\}}\right\}, \tag{4}$$

where $\widehat{\delta}_i$ is obtained through constrained projected-gradient search. Here, $(\eta > 0)$ is a small constant for numerical stability. A larger score indicates that a greater relative reduction in denoising loss is accessible within the prescribed neighborhood.

Recovery accessibility and reappearance. We next examine whether recovery accessibility measured immediately after deletion is associated with subsequent reappearance. To exclude immediate deletion failures, we restrict the analysis to targets satisfying $s_i(\theta_i) < \tau$ and track their outcomes over the following 30 deletion stages. As shown in Figure 2, targets that later reappear consistently exhibit higher recovery accessibility than those that remain forgotten across different unlearning methods.

Table 1 further reports the Spearman correlation between recovery accessibility and maximum future SSCD. Since Immediate SSCD may influence both quantities, we also report partial Spearman correlation controlling for Immediate SSCD. We additionally report AUROC to evaluate how well recovery accessibility distinguishes reappearing targets from those that remain forgotten. Across methods, recovery accessibility consistently shows positive correlation with maximum future SSCD, and the association remains positive after controlling for Immediate SSCD. The AUROC values further indicate meaningful predictive ability for subsequent reappearance.

These findings show that immediate forgetting alone does not fully determine whether a target will remain forgotten. Even when the target satisfies the immediate SSCD criterion, nearby representations may remain easy for the model to denoise. This residual local recoverability is associated with later reappearance and motivates explicitly targeting the most recoverable local representation during deletion, forming the basis of LASTING in Section 4.

4 PROPOSED METHOD

We propose *Low-loss Adaptive Search and Targeted Intervention in Neighborhood Geometry* (LASTING). At each optimization step, LASTING finds a low-loss neighbor of the deletion target and uses it to guide the base unlearning objective toward the model's strongest remaining recovery route.

4.1 ADAPTIVE LOW-LOSS NEIGHBOR SEARCH

Let $\bar{\alpha}_t$ denote the cumulative diffusion noise schedule and let $\xi = (t, \varepsilon)$ contain a timestep t and Gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$. For an image $a \in \mathcal{X}$, the noisy input and its standard DDPM denoising loss (Ho et al., 2020) are

$$z_t(a;\xi) = \sqrt{\bar{\alpha}_t}\, a + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \quad \ell_{\text{diff}}(\theta; a, \xi) = \frac{1}{d} \left\| \varepsilon_\theta \left(z_t(a;\xi), t \right) - \varepsilon \right\|_2^2, \tag{5}$$

where d is the dimensionality of the noise prediction. A low value of ℓ_{diff} indicates that the model can accurately denoise the corresponding representation, suggesting that substantial local recovery ability remains.

For target a_i, we define the feasible neighborhood

$$\mathcal{D}_i = \{\delta : \|\delta\|_\infty \le \epsilon_{\text{nbr}},\ a_i + \delta \in \mathcal{X}\}, \tag{6}$$

where ϵ_{nbr} controls the search radius and $\mathcal{X}$ denotes the valid normalized image domain. At iteration q, we sample $\xi_{i,q}$ and define

$$f_{i,q}(\delta) = \ell_{\text{diff}} \left(\theta_{i,q}; a_i + \delta, \xi_{i,q} \right). \tag{7}$$

Starting from $\delta_{i,q}^{(0)} = 0$, we perform J projected sign-gradient descent steps:

$$\delta_{i,q}^{(r+1)} = \Pi_{\mathcal{D}_i} \left[\delta_{i,q}^{(r)} - \eta_{\text{in}} \operatorname{sign} \left(\nabla_\delta f_{i,q} \left(\delta_{i,q}^{(r)} \right) \right) \right], \quad r = 0, \ldots, J-1. \tag{8}$$

The resulting deletion representation is

$$\widetilde{a}_{i,q} = \operatorname{sg} \left(a_i + \delta_{i,q}^{(J)} \right). \tag{9}$$

Here, η_{in} is the inner-search step size, $\Pi_{\mathcal{D}_i}$ enforces the perturbation constraint and valid image range, and sg denotes stop-gradient. We fix $\xi_{i,q}$ during the inner search and reuse it for the subsequent deletion update so that both steps use the same stochastic diffusion state.

4.2 RECOVERY-GUIDED UNLEARNING

Let $\mathcal{L}_{\text{U}}$ denote a base unlearning objective. LASTING replaces the original target a_i with the selected neighbor $\widetilde{a}_{i,q}$ while preserving the remaining components of the base method:

$$\mathcal{L}_{i,q}^{\text{LASTING}} = \mathcal{L}_{\text{U}} \left(\theta_{i,q}; \mathcal{R}_{i,q}, \widetilde{a}_{i,q}, \xi_{i,q} \right). \tag{10}$$

Here, $\mathcal{R}_{i,q}$ is the retain mini-batch and $\xi_{i,q}$ is the sampled diffusion state. The model is updated using the original optimization rule, so LASTING changes only the representation used by the deletion objective.

In our main implementation, we instantiate LASTING on SISS (Alberti et al., 2025). The retain construction, defensive mixture, importance weighting, gradient normalization, clipping, and optimizer remain unchanged. Let $g_{i,q}^R$ and $g_{i,q}^F$ denote the retain and forget gradient components of SISS, respectively, with $g_{i,q}^F$ evaluated at the selected low-loss neighbor $\widetilde{a}_{i,q}$. The resulting LASTING gradient is

$$g_{i,q}^{\text{LASTING}} = g_{i,q}^R - \kappa \frac{g_{i,q}^F}{\left\| g_{i,q}^F \right\|_2}. \tag{11}$$

Then the model update is

$$\theta_{i,q+1} = \mathrm{OptStep}_{\mathrm{SISS}}\left(\theta_{i,q}; g_{i,q}^{\mathrm{LASTING}}\right). \tag{12}$$

Here, κ is the target norm of the deletion-gradient component, and $\mathrm{OptStep}_{\mathrm{SISS}}$ denotes the original SISS optimizer step, including gradient clipping. Thus, LASTING modifies only the representation used to compute the forget-side gradient, while preserving the retain-side computation and the original SISS optimization procedure. This formulation allows LASTING to retain the original SISS update structure while incorporating the adaptively selected low-loss neighbor.

The selected neighbor $\widetilde{a}_{i,q}$ is detached before evaluating the outer objective. Therefore, the outer update does not differentiate through the neighbor search and requires no second-order derivatives. After Q optimization iterations, we set $\theta_i = \theta_{i,Q}$ and proceed to the next deletion request.

5 Experiments

We now evaluate whether LASTING reduces recovery accessibility and improves forgetting persistence under sequential deletion while maintaining generation quality on CelebA-HQ and Stable Diffusion.

Compared methods. We compare LASTING with SISS, EraseDiff, ReTrack, Forward KL-Constrained Unlearning, and Prompt-Free Instance Unlearning (Alberti et al., 2025; Wu et al., 2025; Shi et al., 2026; Khalafi et al., 2026; Lee et al., 2026b). The target order, retain set, model initialization, and evaluation protocol are fixed across methods, while each baseline uses its recommended optimization configuration and update budget. Prompt-Free additionally uses a generative model to produce edited surrogate targets.

5.1 Sequential Unlearning on CelebA-HQ

Experimental setup. We use CelebA-HQ (Karras et al., 2018) and a pretrained 256×256 unconditional DDPM (Ho et al., 2020). We process $K \in \{50, 100, 200, 300\}$ deletion targets sequentially with the same model and fix the target order and retain set across methods. Optimization details are provided in the appendix.

Evaluation metrics. We report Immediate and Final SSCD (Pizzi et al., 2022), Final Success and Final Reappearance from Eq. 2, and FID (Heusel et al., 2017). Lower SSCD and FID indicate stronger forgetting and better generation quality, respectively. We use the forgetting threshold $\tau = 0.7$.

Main comparison. Table 2 compares methods with the deletion target set and order fixed.

At $K = 300$, LASTING achieves 96.78% Final Success and 3.12% Final Reappearance, compared with 91.67% and 7.93% for SISS. EraseDiff and ReTrack have Final Reappearance rates of 53.89% and 49.84%, respectively. LASTING also lowers Final SSCD relative to SISS (0.4034 versus 0.4243), with a FID of 23.14.

The advantages in Final Success and Final Reappearance extend across the evaluated sequence lengths and training seeds. Mean $\pm$ sample standard deviation across training seeds is reported in Appendix A.1. Figure 1b shows how reappearance develops during the reference deletion sequence. A more detailed trajectory-level analysis, including transient and ever reappearance, is provided in Appendix A.3.

Different deletion target sets. The separate target-set comparison in Table 3 further shows that the improvement persists when the deletion targets are changed at $K = 200$.

5.2 Extension to Text-Conditioned Latent Diffusion

Experimental setup and metrics. We further evaluate LASTING on Stable Diffusion v1.4, a text-conditioned latent diffusion model (Rombach et al., 2022). We construct a sequence of $K_{\mathrm{SD}} =$

Table 2: Sequential data-point unlearning on CelebA-HQ for K deletion requests (% of data). Immediate SSCD is measured after each deletion, and final metrics at the last checkpoint. Final Success uses all K targets, while Final Reappearance uses immediately forgotten targets. Results are means over three training seeds with a fixed target set and order.

K (% data)	Method	Immediate SSCD ↓	Final SSCD ↓	Final Success (%)↑	Final Reapp. (%)↓	FID ↓
50 (0.17%)	SISS	0.3661	0.4704	98.00%	2.00%	21.66
	EraseDiff	**0.1759**	**0.3522**	86.67%	12.30%	73.76
	ReTrack	0.3498	0.4599	92.00%	2.21%	24.92
	FU	0.2673	0.3575	99.33%	0.67%	34.74
	Prompt-Free	0.4509	0.6159	77.33%	22.67%	**19.71**
	LASTING	0.3087	0.4016	**100.00%**	**0.00%**	21.37
100 (0.33%)	SISS	0.3465	0.4530	96.33%	3.67%	**18.91**
	EraseDiff	**0.1814**	0.4684	72.67%	26.24%	152.08
	ReTrack	0.3270	0.4481	89.00%	3.29%	24.50
	FU	0.2750	**0.3972**	96.67%	3.33%	34.32
	Prompt-Free	0.4225	0.6124	75.33%	24.67%	19.59
	LASTING	0.2906	0.3994	**98.67%**	**1.33%**	20.79
200 (0.67%)	SISS	0.3153	0.4244	95.67%	4.33%	21.60
	EraseDiff	**0.1886**	0.6356	47.33%	51.99%	168.25
	ReTrack	0.3351	0.5254	80.33%	10.23%	21.76
	FU	0.2560	**0.3652**	87.83%	12.06%	36.92
	Prompt-Free	0.3990	0.6554	55.83%	44.17%	20.66
	LASTING	0.2651	0.3971	**96.83%**	**3.01%**	**19.61**
300 (1.00%)	SISS	0.2917	0.4243	91.67%	7.93%	**21.76**
	EraseDiff	**0.1863**	0.6322	45.44%	53.89%	134.95
	ReTrack	0.3517	0.6739	43.11%	49.84%	28.30
	FU	0.2240	**0.3305**	80.89%	19.06%	33.36
	Prompt-Free	0.3831	0.6751	47.67%	52.33%	21.91
	LASTING	0.2365	0.4034	**96.78%**	**3.12%**	23.14

Table 3: Performance across three deletion configurations on CelebA-HQ at $K = 200$. $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ start at indices 10,000, 15,000, and 20,000. Succ. denotes Final Success over all 200 targets, and Reapp. denotes Final Reappearance among targets forgotten immediately after deletion. The last column gives the mean and sample SD of Reapp. across the three configurations.

	$\mathcal{T}_1$ (Ref.)		$\mathcal{T}_2$		$\mathcal{T}_3$		Mean ± SD
Method	Succ.	Reapp.	Succ.	Reapp.	Succ.	Reapp.	Reapp.
SISS	94.50%	5.50%	79.50%	20.50%	85.50%	14.50%	(13.50±7.55)%
EraseDiff	63.00%	36.68%	92.50%	7.22%	58.00%	40.82%	(28.24±18.32)%
ReTrack	84.50%	8.43%	83.00%	12.83%	68.50%	20.12%	(13.79±5.90)%
FU	65.50%	34.17%	98.50%	1.01%	87.00%	13.00%	(16.06±16.79)%
Prompt-Free	63.00%	37.00%	58.50%	40.91%	57.00%	43.00%	(40.30±3.05)%
LASTING	**97.50%**	**2.50%**	**99.00%**	**1.00%**	**99.00%**	**1.00%**	$\mathbf{(1.50 \pm 0.87)\%}$

30 memorized deletion targets using the prompts identified by Webster (2023). Each target consists of a reference memorized image and its original and modified evaluation prompts. Following Alberti et al. (2025), we generate 128 samples for each modified prompt to construct memorized and non-memorized sets. At $K \in \{10, 20, 30\}$, we generate 16 images per target–prompt pair with fixed

Table 4: Sequential data-point unlearning on Stable Diffusion. Final Suc. denotes Final Success after K deletion requests. Among cases successfully forgotten immediately after deletion, Reapp. measures reappearance under the final model. IQA denotes CLIP-IQA generation quality.

	$K = 10$			$K = 20$			$K = 30$		
Method	Final Suc. ↑	Reapp. ↓	IQA ↑	Final Suc. ↑	Reapp. ↓	IQA ↑	Final Suc. ↑	Reapp. ↓	IQA ↑
SISS	90.00%	**0.00%**	0.48	80.00%	12.12%	0.66	66.67%	22.45%	0.37
EraseDiff	85.00%	14.29%	0.48	75.00%	18.75%	0.55	61.67%	31.25%	0.68
ReTrack	**100.00%**	**0.00%**	**0.71**	92.50%	7.50%	**0.78**	**93.33%**	6.90%	0.62
FU	95.00%	**0.00%**	0.58	**97.50%**	**2.70%**	0.74	61.67%	31.25%	**0.84**
Prompt-Free	75.00%	11.76%	0.56	70.00%	15.63%	0.74	81.67%	10.20%	0.68
LASTING	**100.00%**	**0.00%**	0.48	92.50%	**2.70%**	0.62	88.33%	**5.88%**	0.66



Figure 3: Qualitative comparison on Stable Diffusion. The leftmost column shows the reference memorized image. The remaining columns correspond to unlearning methods, and rows show generations from the immediate and final model states. Prompt: *Chipotle Bag Essays Earn a Spot in Yale's Rare Book Library*

random seeds and report Final Success, Reappearance Rate, and CLIP-IQA (Wang et al., 2023). Further details are provided in the appendix.

Results. Table 4 shows that LASTING's persistence advantage extends to prompt-conditioned latent diffusion. At $K = 30$, LASTING achieves 88.33% Final Success with only 5.88% reappearance, compared with 66.67%/22.45% for SISS, 61.67%/31.25% for EraseDiff, and 93.33%/6.90% for ReTrack. These results indicate that LASTING preserves forgetting under subsequent deletion updates. LASTING also maintains competitive generation quality. The qualitative examples in Figure 3 show that memorized content remains suppressed without obvious visual degradation in the shown examples.

5.3 Ablation and Analysis

Neighbor construction. Table 5 shows that the direction of neighborhood search is critical. LASTING achieves 2.50% Final Reappearance and 97.50% Final Success, compared with 5.50% for SISS, 23.74% for Random Neighbor, and 37.20% for High-Loss Neighbor. These results demonstrate that explicitly suppressing a locally recoverable low-loss representation improves persistent forgetting, whereas random perturbation and high-loss search degrade performance. This contrast indicates that the benefit comes from searching toward low-loss, locally recoverable regions rather than from generic perturbation of the target neighborhood.

Recovery accessibility across target groups. Table 6 further shows that LASTING reduces mean recovery accessibility from 0.0501 to 0.0309, a 38.35% reduction. The reduction reaches

Table 5: Effect of neighbor construction on sequential forgetting. Random Neighbor uses a norm-matched random perturbation, High-Loss Neighbor performs projected loss maximization, and LASTING performs projected loss minimization. Final Success is measured over all 200 targets, whereas Final Reappearance is measured over targets forgotten immediately after deletion.

Variant	Neighbor construction	Imm. SSCD ↓	Final SSCD ↓	Final Success ↑	Final Reapp. ↓	FID ↓
SISS	Observed target ($\delta = 0$)	0.3024	0.4286	94.50%	5.50%	21.44
Random Neighbor	Norm-matched random perturbation	0.4211	0.5996	75.50%	23.74%	**19.49**
High-Loss Neighbor	Projected loss maximization	0.5082	0.6517	54.00%	37.20%	35.03
LASTING (Low-Loss Neighbor)	Projected loss minimization	**0.2577**	**0.3960**	**97.50%**	**2.50%**	19.68

Table 6: Recovery accessibility by LASTING across targets with different recovery accessibility levels. Reduction denotes the relative decrease from SISS to LASTING.

	N	SISS	LASTING	Reduction
Entire set	300	0.050	0.031	38.35%
Top 25%	75	0.161	0.051	68.39%
Top 10%	30	0.344	0.077	77.72%
Top 5%	15	0.558	0.096	82.83%

Table 7: Compatibility with different unlearning objectives on CelebA-HQ at $K = 200$. Final Reapp. denotes Final Reappearance among targets successfully forgotten immediately after deletion.

Method	Final Reapp.
SISS	5.50%
LASTING + SISS	**2.50%**
FU	34.17%
LASTING + FU	**4.50%**

77.72% and 82.83% for the top 10% and 5% most vulnerable targets, confirming that LASTING suppresses the strongest remaining recovery geometry. Hyperparameter sensitivity and computational overhead are reported in the appendix.

Compatibility with different unlearning objectives. We further apply LASTING to Forward KL-Constrained Unlearning. Table 7 shows that it reduces Final Reappearance from 5.50% to 2.50% for SISS and from 34.17% to 4.50% for FU. The substantial improvement under FU indicates that recovery-guided target selection remains effective across both evaluated unlearning objectives, supporting the view that LASTING's benefit arises from its recovery-guided target modification rather than from the specific SISS objective.

6 Conclusion

In this paper, we identify *sequential reappearance*, a failure mode in which a diffusion training instance that is forgotten immediately after deletion becomes recoverable again after subsequent unlearning updates. Our analysis indicates that this failure is associated with greater local recovery accessibility around the deleted target. Motivated by this finding, we propose LASTING, which adaptively identifies and suppresses locally recoverable representations during unlearning. Across diffusion settings, LASTING substantially improves forgetting persistence while maintaining competitive generation quality. These results highlight the need to evaluate diffusion unlearning beyond immediate deletion success and to account for the stability of forgetting under subsequent requests.

AI USE STATEMENT

In this work, we used generative AI tools for retrieval and discovery, including identifying related work and suggesting search queries and keywords. All retrieved sources were manually reviewed by the authors, who determined the final selection and citation of related work.

REFERENCES

Silas Alberti, Kenan Hasanaliyev, Manav Shah, and Stefano Ermon. Data unlearning in diffusion models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 3084–3100, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/08a362bd4ae1934e099ce025f06039fe-Paper-Conference.pdf.

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.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1019–1028. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/dinh17b.html.

Ali Ebrahimpour-Boroojeny, Hari Sundaram, and Varun Chandrasekaran. Not all wrong is bad: Using adversarial examples for unlearning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14950–14971. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/ebrahimpour-boroojeny25a.html.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 53643–53673, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/ec4d2e436794d1bf55ca83f5ebb31887-Paper-Conference.pdf.

Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards LLM unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15762–15778. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/fan25e.html.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2426–2436, October 2023.

Hongcheng Gao, Tianyu Pang, Chao Du, Taihang Hu, Zhijie Deng, and Min Lin. Meta-unlearning on diffusion models: Preventing relearning unlearned concepts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2131–2141, October 2025.

Naveen George, Karthik Nandan Dasaraju, Rutheesh Reddy Chittepu, and Konda Reddy Mopuri. The illusion of unlearning: The unstable nature of machine unlearning in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13393–13402, June 2025.

Naveen George, Naoki Murata, Yuhta Takida, Konda Reddy Mopuri, and Yuki Mitsufuji. Locality-aware continual unlearning for diffusion models. In *European Conference on Computer Vision (ECCV)*. Springer, 2026.

Hyeonggeun Han, Sehwan Kim, Hyungjun Joo, Sangwoo Hong, and Jungwoo Lee. Adjusting initial noise to mitigate memorization in text-to-image diffusion models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 684–711. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/012fb887ac427dea14cd2f413230f5f7-Paper-Conference.pdf.

Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 17170–17194. Curran Associates, Inc., 2023. doi: 10.52202/075280-0751. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/376276a95781fa17c177b1ccdd0a03ac-Paper-Conference.pdf.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

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

Hsiang Hsu, Pradeep Niroula, Zichang He, Ivan Brugere, Freddy Lecue, and Richard Chen. The unseen threat: Residual knowledge in machine unlearning under perturbed samples. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 26721–26763. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2656bba937d78593fbd99ace9f14e311-Paper-Conference.pdf.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

Shervin Khalafi, Alejandro Ribeiro, and Dongsheng Ding. Unlearning in diffusion models: A unified framework with KL divergence and likelihood constraints. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=98xDF2hkIy.

Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22691–22702, October 2023.

Justin Lee, Zheda Mai, Jinsu Yoo, Chongyu Fan, Cheng Zhang, and Wei-Lun Chao. Continual unlearning for text-to-image diffusion models: A regularization perspective. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=BsY20r9FOM.

Kyungryeol Lee, Kyeonghyun Lee, Seongmin Hong, Byung Hyun Lee, and Se Young Chun. Unlearning the unpromptable: Prompt-free instance unlearning in diffusion models, 2026b. URL https://arxiv.org/abs/2603.10445. Accepted to the Non-Archival Track of the CVPR 2026 Workshop on MUV.

Gen Li, Yang Xiao, Jie Ji, Kaiyuan Deng, Bo Hui, Linke Guo, and Xiaolong Ma. Sculpting memory: Multi-concept forgetting in diffusion models via dynamic mask and concept-aware optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19659–19668, October 2025.

Shilin Lu, Zilan Wang, Leyang Li, Yanzhu Liu, and Adams Wai-Kin Kong. Mace: Mass concept erasure in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6430–6440, June 2024.

Saber Malekmohammadi, Hong kyu Lee, and Li Xiong. Sharpness-aware parameter selection for machine unlearning, 2025. URL https://arxiv.org/abs/2504.06398.

Claudia Merger and Sebastian Goldt. Local coverage governs memorization in diffusion models. In *The Second Workshop on the Impact of Memorization on Trustworthy Foundation Models at ICML*, 2026. URL https://openreview.net/forum?id=wUCcl1vNxd.

Eun-Ju Park, Youjin Shin, and Simon S. Woo. Robust continual unlearning against knowledge erosion and forgetting reversal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 7978–7987, June 2026.

Jinseong Park and Mijung Park. Not every time and frequency need to be forgotten in diffusion unlearning. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026. URL https://openreview.net/forum?id=ONZmXOH029.

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.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Qitan Shi, Cheng Jin, Jiawei Zhang, and Yuantao Gu. Retrack: Data unlearning in diffusion models through redirecting the denoising trajectory. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=oTUZfa0iPv.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6048–6058, June 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 47783–47803. Curran Associates, Inc., 2023b. doi: 10.52202/075280-2071. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/9521b6e7f33e039e7d92e23f5e37bbf4-Paper-Conference.pdf.

Vinith Menon Suriyakumar, Rohan Alur, Ayush Sekhari, Manish Raghavan, and Ashia C. Wilson. Unstable unlearning: The hidden risk of concept resurgence in diffusion models. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=Vj0Z2wspQ5.

Haoran Tang and Rajiv Khanna. Sharpness-aware machine unlearning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=BZ8I2tXomt.

Kartik Thakral, Tamar Glaser, Tal Hassner, Mayank Vatsa, and Richa Singh. Continual unlearning for foundational text-to-image models without generalization erosion, 2025. URL https://arxiv.org/abs/2503.13769.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring CLIP for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 2555–2563, 2023.

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 B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 35647–35662, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/993760d9b1d63a385611c7c9ef8f3ecb-Paper-Conference.pdf.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 2958–2969. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1ef91c212e30e14bf125e9374262401f-Paper.pdf.

Jing Wu, Trung Le, Munawar Hayat, and Mehrtash Harandi. Erasing undesirable influence in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28263–28273, June 2025.

Gong Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1755–1764, June 2024a.

Yihua Zhang, Chongyu Fan, Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jiancheng Liu, Gaoyuan Zhang, Gaowen Liu, Ramana Kompella, Xiaoming Liu, and Sijia Liu. Unlearncanvas: Stylized image dataset for enhanced machine unlearning evaluation in diffusion models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 96387–96423. Curran Associates, Inc., 2024b. doi: 10.52202/079017-3055. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/aebf4822d30c3f2600566af7eba83548-Paper-Datasets_and_Benchmarks_Track.pdf.

Yimeng Zhang, Xin Chen, Jinghan Jia, Yihua Zhang, Chongyu Fan, Jiancheng Liu, Mingyi Hong, Ke Ding, and Sijia Liu. Defensive unlearning with adversarial training for robust concept erasure in diffusion models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 36748–36776. Curran Associates, Inc., 2024c. doi: 10.52202/079017-1158. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/40954ac18a457dd5f11145bae6454cdf-Paper-Conference.pdf.

SUPPLEMENTARY MATERIAL CONTENTS

A. Additional Experimental Analyses

A.1 Results Across Training Seeds

A.2 Comparison at Matched Computation

A.3 Detailed Analysis of Reappearance

B. Sensitivity Analyses

B.1 Number of Inner-Search Steps

B.2 Neighborhood Radius

B.3 Evaluation Timestep

C. Implementation Details

C.1 CelebA-HQ

C.2 Stable Diffusion

D. Additional Qualitative Results

D.1 CelebA-HQ

D.2 Stable Diffusion

A ADDITIONAL EXPERIMENTAL ANALYSES

A.1 RESULTS ACROSS TRAINING SEEDS

We repeat the CelebA-HQ sequential unlearning experiments with training seeds 0, 42, and 10000 at $K \in \{50, 100, 200, 300\}$. For each sequence length, the deletion targets, their order, and the evaluation protocol are fixed across seeds, and each method uses the same optimization settings in all three runs. Table 8 reports the resulting mean $\pm$ sample standard deviation.

A.2 COMPARISON AT MATCHED COMPUTATION

Computational overhead. We compare the computational overhead of LASTING with SISS, EraseDiff, ReTrack, Forward KL-Constrained Unlearning (FU), and Prompt-Free Instance Unlearning. Each method uses its reported CelebA-HQ configuration and update budget. We measure per-target wall-clock time and peak process GPU memory on a single NVIDIA RTX PRO 6000 Blackwell Server Edition GPU. As shown in Table 9, LASTING takes 960.43 seconds per target due to its inner search. Its peak GPU memory is 13.63 GiB, the lowest among the evaluated methods.

Increased-budget SISS control. We further evaluate whether additional SISS optimization reproduces the persistence achieved by recovery-guided target replacement. To approximately match the computational budget of LASTING, we increase the number of SISS training steps per deletion target based on the measured per-target computation time of standard SISS and LASTING. Table 10 compares the reference SISS run, the increased-budget SISS run, and LASTING along the deletion sequence.

The increased-budget SISS run obtains the lowest Immediate SSCD at every reported checkpoint, showing stronger forgetting immediately after each deletion. Its Final Reappearance nevertheless reaches 29.50% at $K = 200$ and 31.00% at $K = 300$, compared with 5.50% and 9.73% for reference SISS. LASTING reduces these rates to 2.50% and 2.33%, respectively. At $K = 300$, it also maintains 97.67% Final Success and a Final SSCD of 0.3917, compared with 69.00% and 0.5377 for increased-budget SISS. Increasing the SISS optimization budget therefore strengthens immediate forgetting without reproducing the persistence obtained by LASTING.

Table 8: Sequential data-point unlearning on CelebA-HQ for K deletion requests (% of data). Immediate SSCD is measured after each deletion, and final metrics at the last checkpoint. Final Success uses all K targets, while Final Reappearance uses immediately forgotten targets. Results are reported as mean $\pm$ sample standard deviation over three training seeds with a fixed target set and order. Standard deviations for percentage metrics are expressed in percentage points.

K (% data)	Method	Immediate SSCD ↓	Final SSCD ↓	Final Success (%)↑	Final Reapp. (%)↓	FID ↓
50 (0.17%)	SISS	0.3661 ± 0.0102	0.4704 ± 0.0204	98.00 ± 2.00	2.00 ± 2.00	21.66 ± 3.11
	EraseDiff	$\mathbf{0.1759 \pm 0.0125}$	$\mathbf{0.3522 \pm 0.0895}$	86.67 ± 10.07	12.30 ± 8.38	73.76 ± 29.75
	ReTrack	0.3498 ± 0.0050	0.4599 ± 0.0178	92.00 ± 2.00	2.21 ± 2.13	24.92 ± 5.04
	FU	0.2673 ± 0.0358	0.3575 ± 0.0096	99.33 ± 1.15	0.67 ± 1.15	34.74 ± 11.17
	Prompt-Free	0.4509 ± 0.0538	0.6159 ± 0.0377	77.33 ± 11.02	22.67 ± 11.02	$\mathbf{19.71 \pm 2.00}$
	LASTING	0.3087 ± 0.0113	0.4016 ± 0.0117	$\mathbf{100.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	21.37 ± 4.88
100 (0.33%)	SISS	0.3465 ± 0.0086	0.4530 ± 0.0385	96.33 ± 0.58	3.67 ± 0.58	$\mathbf{18.91 \pm 2.30}$
	EraseDiff	$\mathbf{0.1814 \pm 0.0185}$	0.4684 ± 0.1092	72.67 ± 15.95	26.24 ± 15.05	152.08 ± 81.70
	ReTrack	0.3270 ± 0.0107	0.4481 ± 0.0247	89.00 ± 1.73	3.29 ± 2.19	24.50 ± 3.49
	FU	0.2750 ± 0.0257	$\mathbf{0.3972 \pm 0.0269}$	96.67 ± 4.16	3.33 ± 4.16	34.32 ± 19.85
	Prompt-Free	0.4225 ± 0.0417	0.6124 ± 0.0404	75.33 ± 11.72	24.67 ± 11.72	19.59 ± 2.14
	LASTING	0.2906 ± 0.0113	0.3994 ± 0.0193	$\mathbf{98.67 \pm 0.58}$	$\mathbf{1.33 \pm 0.58}$	20.79 ± 3.16
200 (0.67%)	SISS	0.3153 ± 0.0113	0.4244 ± 0.0154	95.67 ± 1.04	4.33 ± 1.04	21.60 ± 0.66
	EraseDiff	$\mathbf{0.1886 \pm 0.0212}$	0.6356 ± 0.0337	47.33 ± 13.65	51.99 ± 13.36	168.25 ± 83.64
	ReTrack	0.3351 ± 0.0038	0.5254 ± 0.0102	80.33 ± 4.01	10.23 ± 2.03	21.76 ± 5.98
	FU	0.2560 ± 0.0064	$\mathbf{0.3652 \pm 0.2257}$	87.83 ± 19.37	12.06 ± 19.17	36.92 ± 14.39
	Prompt-Free	0.3990 ± 0.0352	0.6554 ± 0.0269	55.83 ± 6.79	44.17 ± 6.79	20.66 ± 0.76
	LASTING	0.2651 ± 0.0081	0.3971 ± 0.0064	$\mathbf{96.83 \pm 0.58}$	$\mathbf{3.01 \pm 0.50}$	$\mathbf{19.61 \pm 0.49}$
300 (1.00%)	SISS	0.2917 ± 0.0097	0.4243 ± 0.0043	91.67 ± 1.76	7.93 ± 1.60	$\mathbf{21.76 \pm 5.49}$
	EraseDiff	$\mathbf{0.1863 \pm 0.0181}$	0.6322 ± 0.0195	45.44 ± 3.56	53.89 ± 3.78	134.95 ± 53.63
	ReTrack	0.3517 ± 0.0082	0.6739 ± 0.0093	43.11 ± 3.67	49.84 ± 5.79	28.30 ± 4.47
	FU	0.2240 ± 0.0245	$\mathbf{0.3305 \pm 0.2940}$	80.89 ± 31.66	19.06 ± 31.58	33.36 ± 16.06
	Prompt-Free	0.3831 ± 0.0317	0.6751 ± 0.0226	47.67 ± 6.49	52.33 ± 6.49	21.91 ± 1.96
	LASTING	0.2365 ± 0.0089	0.4034 ± 0.0114	$\mathbf{96.78 \pm 1.02}$	$\mathbf{3.12 \pm 1.08}$	23.14 ± 5.41

Table 9: Computational overhead on CelebA-HQ. Each method uses its reported optimization configuration and update budget.

Method	Time per target (s) ↓	Peak process GPU memory (GiB) ↓
SISS	257.70	13.83
EraseDiff	252.92	21.14
ReTrack	222.29	19.30
FU	215.14	19.51
Prompt-Free	281.33	21.56
LASTING	960.43	13.63

A.3 Detailed Analysis of Reappearance

We further analyze reappearance over deletion trajectories of $K \in \{50, 100, 200, 300\}$. For each target satisfying $s_i(\theta_i) < \tau$ immediately after deletion, we examine all subsequent checkpoints using the same SSCD threshold $\tau = 0.7$ as in the main paper. *No Reappearance* indicates that the target remains below the threshold at all subsequent checkpoints, whereas *Transient Reappearance* indicates that it reaches or exceeds the threshold but returns below it at the final checkpoint. *Final Reappearance* indicates that the target is at or above the threshold under the final model. Ever Reappearance consists of transient and final reappearance.

Table 10: Sequential unlearning results for reference SISS, increased-budget SISS, and LASTING on the recorded CelebA-HQ deletion sequence. These are single-run results. Final Success is measured over all K targets. Final Reappearance is measured among targets successfully forgotten immediately after deletion.

K	Method	Immediate SSCD ↓	Final SSCD ↓	Final Success (%) ↑	Final Reappearance ↓
50	SISS	0.3545	0.4739	100.00	0/50 (0.00%)
	SISS (extended)	0.1539	0.3819	96.00	2/50 (4.00%)
	LASTING	0.2965	0.3944	100.00	0/50 (0.00%)
100	SISS	0.3366	0.4171	96.00	4/100 (4.00%)
	SISS (extended)	0.1161	0.5279	89.00	11/100 (11.00%)
	LASTING	0.2807	0.3785	99.00	1/100 (1.00%)
200	SISS	0.3024	0.4286	94.50	11/200 (5.50%)
	SISS (extended)	0.0851	0.5691	70.50	59/200 (29.50%)
	LASTING	0.2577	0.3960	97.50	5/200 (2.50%)
300	SISS	0.2807	0.4238	89.67	29/298 (9.73%)
	SISS (extended)	0.0664	0.5377	69.00	93/300 (31.00%)
	LASTING	0.2281	0.3917	97.67	7/300 (2.33%)

As shown in Table 11, LASTING achieves the lowest Ever Reappearance rate across all four sequence lengths. At $K = 50$, none of its immediately forgotten targets reappear. At $K = 100$ and $K = 200$, its Ever Reappearance rates are 4.00% and 4.50%, compared with 9.00% and 13.50% for SISS. At $K = 300$, LASTING maintains the highest No Reappearance rate of 93.00% and the lowest Ever Reappearance rate of 7.00%. Its 21 reappearing targets consist of 14 transient and 7 final cases. In comparison, SISS exhibits an Ever Reappearance rate of 23.49%, while the other baselines range from 65.02% to 80.81%. These results show that LASTING reduces both transient and final reappearance throughout the deletion trajectory.

Subsequent deletion updates can also correct some immediate forgetting failures. Our reappearance metrics instead focus on cases where a successfully forgotten target later becomes memorized again, because this represents the loss of an already achieved forgetting state.

B Sensitivity Analyses

We conduct ablations on the number of inner-search steps J and the neighborhood radius ϵ_{nbr} in LASTING. All hyperparameter ablations use the CelebA-HQ sequential deletion setting with SISS as the outer unlearning objective. Within each ablation, we vary only the component under study while keeping the deletion order, retain set, outer-update budget, optimizer, and evaluation protocol fixed.

We follow the persistence metrics defined in the main paper using SSCD similarity with the operational threshold $\tau = 0.7$. Final Success is normalized by all K deletion targets, whereas Final Reappearance is normalized by the immediate-success cohort. We report each rate as both a raw count and a percentage. Immediate and Final SSCD are averaged over all K targets.

B.1 Number of Inner-Search Steps

We examine how the inner-search budget affects forgetting persistence by varying $J \in \{1, 4, 8, 12\}$ on the same $K = 300$ deletion sequence. All configurations use $\epsilon_{\text{nbr}} = 4/255$ and share the same outer optimization and evaluation settings. The diffusion timestep and noise are fixed throughout each inner search and reused in the subsequent outer update.

As shown in Table 12, $J = 8$ achieves the lowest Final Reappearance rate of 2.33% and the lowest Final SSCD of 0.3917. Increasing the search budget from $J = 4$ to $J = 8$ improves both metrics,

Table 11: Trajectory-level reappearance on CelebA-HQ for training seed 42. Immediate Success is reported over all K targets. The remaining categories are reported over targets successfully forgotten immediately after deletion. Best results for each K are shown in bold.

K	Method	Immediate Success ↑	No Reapp. ↑	Transient Reapp. ↓	Final Reapp. ↓	Ever Reapp. ↓
50	SISS	**50/50 (100.00%)**	49/50 (98.00%)	1/50 (2.00%)	**0/50 (0.00%)**	1/50 (2.00%)
	EraseDiff	**50/50 (100.00%)**	33/50 (66.00%)	11/50 (22.00%)	6/50 (12.00%)	17/50 (34.00%)
	ReTrack	47/50 (94.00%)	45/47 (95.74%)	**0/47 (0.00%)**	2/47 (4.26%)	2/47 (4.26%)
	FU	**50/50 (100.00%)**	48/50 (96.00%)	2/50 (4.00%)	**0/50 (0.00%)**	2/50 (4.00%)
	Prompt-Free	**50/50 (100.00%)**	43/50 (86.00%)	2/50 (4.00%)	5/50 (10.00%)	7/50 (14.00%)
	LASTING	**50/50 (100.00%)**	**50/50 (100.00%)**	**0/50 (0.00%)**	**0/50 (0.00%)**	**0/50 (0.00%)**
100	SISS	**100/100 (100.00%)**	91/100 (91.00%)	5/100 (5.00%)	4/100 (4.00%)	9/100 (9.00%)
	EraseDiff	**100/100 (100.00%)**	59/100 (59.00%)	32/100 (32.00%)	9/100 (9.00%)	41/100 (41.00%)
	ReTrack	92/100 (92.00%)	87/92 (94.57%)	**2/92 (2.17%)**	3/92 (3.26%)	5/92 (5.43%)
	FU	**100/100 (100.00%)**	82/100 (82.00%)	10/100 (10.00%)	8/100 (8.00%)	18/100 (18.00%)
	Prompt-Free	**100/100 (100.00%)**	60/100 (60.00%)	24/100 (24.00%)	16/100 (16.00%)	40/100 (40.00%)
	LASTING	**100/100 (100.00%)**	**96/100 (96.00%)**	3/100 (3.00%)	**1/100 (1.00%)**	**4/100 (4.00%)**
200	SISS	**200/200 (100.00%)**	173/200 (86.50%)	16/200 (8.00%)	11/200 (5.50%)	27/200 (13.50%)
	EraseDiff	199/200 (99.50%)	78/199 (39.20%)	48/199 (24.12%)	73/199 (36.68%)	121/199 (60.80%)
	ReTrack	178/200 (89.00%)	126/178 (70.79%)	37/178 (20.79%)	15/178 (8.43%)	52/178 (29.21%)
	FU	199/200 (99.50%)	72/199 (36.18%)	59/199 (29.65%)	68/199 (34.17%)	127/199 (63.82%)
	Prompt-Free	**200/200 (100.00%)**	80/200 (40.00%)	46/200 (23.00%)	74/200 (37.00%)	120/200 (60.00%)
	LASTING	**200/200 (100.00%)**	**191/200 (95.50%)**	**4/200 (2.00%)**	**5/200 (2.50%)**	**9/200 (4.50%)**
300	SISS	298/300 (99.33%)	228/298 (76.51%)	41/298 (13.76%)	29/298 (9.73%)	70/298 (23.49%)
	EraseDiff	297/300 (99.00%)	57/297 (19.19%)	67/297 (22.56%)	173/297 (58.25%)	240/297 (80.81%)
	ReTrack	263/300 (87.67%)	92/263 (34.98%)	23/263 (8.75%)	148/263 (56.27%)	171/263 (65.02%)
	FU	299/300 (99.67%)	88/299 (29.43%)	45/299 (15.05%)	166/299 (55.52%)	211/299 (70.57%)
	Prompt-Free	**300/300 (100.00%)**	94/300 (31.33%)	71/300 (23.67%)	135/300 (45.00%)	206/300 (68.67%)
	LASTING	**300/300 (100.00%)**	**279/300 (93.00%)**	**14/300 (4.67%)**	**7/300 (2.33%)**	**21/300 (7.00%)**

Table 12: Sensitivity to the number of inner-search steps.

J	Time per Target (s) ↓	Immediate SSCD ↓	Final SSCD ↓	Final Success ↑	Final Reapp. ↓
1	419.31	0.3142	0.5141	193/300 (64.33%)	90/283 (31.80%)
4	647.26	0.2382	0.4205	287/300 (95.67%)	13/300 (4.33%)
8	960.43	**0.2281**	**0.3917**	**293/300 (97.67%)**	**7/300 (2.33%)**
12	1301.56	0.2932	0.4580	284/300 (94.67%)	13/297 (4.38%)

whereas $J = 12$ provides no further benefit and increases Final SSCD and endpoint reappearance. We therefore use $J = 8$ for the CelebA-HQ experiments.

B.2 Neighborhood Radius

Neighborhood radii are reported in the conventional $[0, 1]$ pixel scale. Since the implementation normalizes images to $[-1, 1]$, a radius of $r/255$ corresponds to $2r/255$ in the implementation space. All perturbations are projected onto the corresponding ℓ_∞ neighborhood and the valid normalized image domain.

We examine how the neighborhood size affects forgetting persistence by varying $\epsilon_{\text{nbr}} \in \{4/255, 8/255, 12/255, 16/255\}$ on the same $K = 100$ deletion sequence. All configurations use $J = 4$ inner-search steps and share the same outer optimization and evaluation settings. The reported radii follow the conventional $[0, 1]$ pixel scale.

Table 13: Sensitivity to the neighborhood radius.

ϵ_{nbr}	Immediate SSCD ↓	Final SSCD ↓	Final Success ↑	Final Reapp. ↓
$4/255$	0.2969	0.4000	99/100 (99.00%)	0/99 (0.00%)
$8/255$	**0.2746**	**0.3495**	**100/100 (100.00%)**	**0/100 (0.00%)**
$12/255$	0.3969	0.5300	81/100 (81.00%)	3/84 (3.57%)
$16/255$	0.5626	0.7030	44/100 (44.00%)	19/63 (30.16%)

As shown in Table 13, $\epsilon_{\text{nbr}} = 8/255$ exhibits no final reappearance and achieves a Final SSCD of 0.3495. Increasing the radius to $12/255$ and $16/255$ raises Final SSCD to 0.5300 and 0.7030, respectively, while also producing more final reappearance. These results indicate that expanding the search region beyond $8/255$ weakens forgetting persistence under the tested setting. Although $8/255$ performs best in this ablation, we use $4/255$ as a conservative default for the longer main experiments.

B.3 Evaluation Timestep

We examine the stability of the evaluation results across two evaluation timesteps, $t_{\text{eval}} \in \{250, 350\}$, on the $K = 300$ deletion sequence. For each timestep, results are aggregated over three recovery-noise seeds.

Table 14: Sensitivity to evaluation timestep at $K = 300$. For each evaluation timestep, results are reported as mean ± sample standard deviation over three recovery-noise seeds. The best result for each evaluation timestep is shown in bold.

Method	t_{eval}	Immediate SSCD ↓	Final SSCD ↓	Final Success (%) ↑	Final Reapp. (%) ↓
SISS	250	0.2831±0.0030	0.4244±0.0032	90.33±0.67	9.16±0.53
	350	0.2401±0.0026	**0.4368±0.0024**	93.44±0.96	6.35±1.14
EraseDiff	250	**0.1791±0.0099**	0.6602±0.0053	40.00±1.33	59.78±1.46
	350	**0.0861±0.0077**	0.5670±0.0080	82.89±1.92	17.11±1.92
ReTrack	250	0.3429±0.0015	0.6858±0.0021	39.22±1.17	56.60±1.12
	350	0.3284±0.0030	0.6343±0.0009	64.44±0.69	31.16±1.28
FU	250	0.2320±0.0034	0.6175±0.0040	62.78±1.35	37.08±1.26
	350	0.2088±0.0045	0.5341±0.0037	94.22±1.39	5.78±1.39
Prompt-Free	250	0.3463±0.0006	0.6516±0.0026	55.11±0.84	44.89±0.84
	350	0.3426±0.0012	0.6249±0.0004	66.22±1.02	33.78±1.02
LASTING	250	0.2299±0.0036	**0.3900±0.0015**	**97.11±0.51**	**2.89±0.51**
	350	0.1920±0.0012	0.4372±0.0029	**97.44±0.51**	**2.56±0.51**

As shown in Table 14, LASTING maintains its forgetting-persistence advantage across recovery-noise seeds and evaluation timesteps. This consistency indicates that the reduction in sequential reappearance achieved by LASTING persists across the evaluated recovery-noise seeds and evaluation settings, while the small standard deviations of LASTING further demonstrate that its performance is stable across the evaluated seeds.

C Implementation Details

We provide implementation and evaluation details for the CelebA-HQ and Stable Diffusion experiments.

C.1 CelebA-HQ

Dataset and sequential deletion protocol. We use the 30,000-image CelebA-HQ dataset and the pretrained `google/ddpm-celebahq-256` unconditional DDPM. The 256×256 images are normalized to $[-1, 1]$. The deletion sequence contains 300 targets, ordered from `10000.jpg` to `10299.jpg`, and we report the prefix checkpoints at $K \in \{50, 100, 200, 300\}$. After unlearning target a_i, the resulting model θ_i initializes deletion stage $i + 1$.

The retain pool is fixed throughout the trajectory. All 300 deletion targets are excluded before sequential unlearning begins, leaving 29,700 retain images. The target order, retain pool, and model initialization are shared across all methods.

For recovery evaluation, each target is noised at diffusion timestep $t = 250$ and reconstructed through the reverse process from $t = 250$ to 0. We use one reconstruction per target and the same evaluation seed and noise realization across all methods, including Prompt-Free Instance Unlearning. Each reconstruction is saved as a PNG and evaluated using the same SSCD encoder and ImageNet preprocessing. We use the memorization threshold $\tau = 0.7$. Immediate SSCD is measured using θ_i, whereas Final SSCD for a prefix of length K is measured using θ_K.

FID is evaluated at each reported prefix checkpoint using 10,000 generated images, 50 DDPM sampling steps, and the full CelebA-HQ reference set.

Compared methods and optimization details. We compare LASTING with SISS, EraseDiff, ReTrack, Forward KL-Constrained Unlearning (FU), and Prompt-Free Instance Unlearning. Each baseline uses the best-performing or recommended optimization configuration from its original work. Table 15 reports the configurations used for the CelebA-HQ experiments.

Table 15: Optimization settings for the CelebA-HQ experiments.

Method	Optimizer	Learning rate	Batch	Method-specific setting
SISS	AdamW	5×10^{-6}	4	$\lambda = 0.5$, `scaling_norm`$= 500$
EraseDiff	AdamW	5×10^{-6}	4	$\eta = 10^{-3}$
ReTrack	AdamW	5×10^{-6}	4	$k = 10$, $\lambda = 0.005$
FU	Adam	5×10^{-6}	4	$\epsilon_{\mathrm{KL}} = 0.01$, $\eta_\lambda = 0.2$
Prompt-Free	AdamW	5×10^{-6}	4	$\beta = 5 \times 10^{-5}$, gradient surgery
LASTING	AdamW	5×10^{-6}	4	$\epsilon_{\mathrm{nbr}} = 4/255$, $J = 8$

All methods use a constant learning-rate schedule without warmup and are trained in FP32. SISS, EraseDiff, ReTrack, and LASTING use a maximum gradient norm of 1.0. FU uses 0.3, and Prompt-Free does not apply gradient clipping. Prompt-Free uses a fixed edited surrogate for each target and applies its original gradient-surgery update.

LASTING implementation. LASTING is implemented on top of SISS while leaving its retain construction and outer deletion objective unchanged. The inner search uses the raw DDPM ϵ-prediction MSE. We set $\epsilon_{\mathrm{nbr}} = 4/255$, $J = 8$, and $\eta_{\mathrm{in}} = \epsilon_{\mathrm{nbr}}/J = 0.5/255$ in the conventional $[0, 1]$ pixel scale. These values correspond to $8/255$ and $1/255$, respectively, in the implementation's $[-1, 1]$ scale.

Each search starts from zero perturbation and performs projected sign-gradient descent without random initialization. After each inner step, the perturbation is projected onto the ℓ_∞ neighborhood, and the resulting image is clipped to $[-1, 1]$. The diffusion timestep and noise are fixed across the J inner steps and reused in the subsequent outer update.

FU+LASTING implementation. We also apply the low-loss neighbor replacement of LASTING to FU. The selected neighbor replaces the observed deletion target in FU's forget-side computation. The retain construction and FU optimization rule remain unchanged. FU and FU+LASTING use the same deletion sequence and evaluation protocol.

Recovery accessibility measurement. We measure recovery accessibility at the checkpoint $\theta_i^{(u)}$ immediately after method u deletes target a_i. For each target, the fixed bank $\mathcal{B}_i$ contains timesteps $\{50, 100, 150, 200, 250\}$ and two Gaussian noise draws per timestep. We use the base seed 20260713 with a target-specific index offset. For each target, the same bank is reused across methods and for both the observed target and all perturbed candidates evaluated during the search.

We set $\epsilon_{\mathrm{acc}} = 0.03137255 \simeq 8/255$ in the normalized $[-1, 1]$ image space, corresponding to $4/255$ in the $[0, 1]$ pixel scale. Starting from zero perturbation, we perform eight projected sign-gradient descent steps with step size $\epsilon_{\mathrm{acc}}/8 \simeq 1/255$. Each step applies ℓ_∞ projection and image-range clipping. We do not use random initialization or multiple restarts.

For the association analysis in the main paper, we retain targets satisfying $s_i(\theta_i^{(u)}) < \tau$. For each eligible target with $i < K$, we measure its maximum future SSCD over the next 30 deletion stages as

$$m_{i,u}^{(30)} = \max_{i<j\le \min(i+30,K)} s_i\left(\theta_j^{(u)}\right). \tag{13}$$

The upper endpoint is truncated at the final checkpoint for targets near the end of the sequence. A target is labeled as reappearing when $m_{i,u}^{(30)} \ge \tau$ and as remaining forgotten otherwise. For the method-level comparison, we average recovery accessibility over the same $K = 300$ deletion targets for SISS and LASTING.

C.2 STABLE DIFFUSION

Data and sequential deletion protocol. We implement Stable Diffusion v1.4 using the Hugging Face Diffusers library. We select $K_{\mathrm{SD}} = 30$ deletion targets from the fully memorized prompt–image pairs identified by Webster and use the data construction of SISS. Each target contains one reference memorized image, its original prompt, a modified prompt, and 128 images generated from the modified prompt.

Following SISS, we use target-specific KMeans classifiers to partition the generated images into memorized and non-memorized subsets. The memorized subset serves as the stage-specific forget set, while the non-memorized subset serves as the retain set. Across the 30 target partitions, this yields 3,840 generated images in total.

We order the 30 targets lexicographically and apply the deletion requests sequentially. The first stage starts from the pretrained Stable Diffusion v1.4 pipeline, and the model obtained at stage i initializes stage $i + 1$. Each stage uses only the forget and retain sets of its current target, while the updated model parameters are carried forward. All methods use the same target order, target assets, initial model, and checkpoint-transfer rule. We report the prefix checkpoints at $K \in \{10, 20, 30\}$.

Memorization and CLIP-IQA evaluation protocol. We evaluate memorization separately for the original and modified prompts and measure generation quality over their combined outputs. The protocol applies the main-paper persistence metrics to target–prompt cases using a target-specific KMeans classifier.

Target-specific memorization classifier. We use the precomputed KMeans classifier stored for each target in the SISS data construction. All evaluation images are generated at 512×512 resolution and passed to the classifier without resizing. Let $h_i(x) = 1$ indicate that the classifier for target i predicts image x as memorized, while $h_i(x) = 0$ denotes a non-memorized image. We keep each classifier fixed across both prompt types, all evaluation checkpoints, and all compared methods.

Prompt-level forgetting. Immediately after the deletion stage, we evaluate the current target using its original and modified prompts. At $K \in \{10, 20, 30\}$, we reevaluate every target processed through stage K. For target i, prompt type $p \in \mathcal{P} = \{\mathrm{original}, \mathrm{modified}\}$, and checkpoint θ_j, we generate images $x^{(j)}_{i,p,r}$ using fixed seeds $r \in \{0, \ldots, 15\}$. All methods use 50 denoising steps and a guidance scale of 7.5. We define the target–prompt success indicator as

$$z^{(j)}_{i,p} = \mathbf{1}\left[\sum_{r=0}^{15} h_i\left(x^{(j)}_{i,p,r}\right) = 0\right]. \tag{14}$$

A target–prompt case is successfully forgotten only when all 16 generations are classified as non-memorized. We apply this condition independently to the two prompt types, giving two evaluation cases for each target.

Target–prompt aggregation. At prefix K, the K processed targets contribute $2K$ target–prompt cases. We apply the main-paper persistence metrics to these cases as

$$\mathrm{FinalSuccess}(K) = \frac{1}{2K}\sum_{i=1}^{K}\sum_{p\in\mathcal{P}} z^{(K)}_{i,p}, \tag{15}$$

$$\mathrm{ReappRate}(K) = \frac{\sum_{i=1}^{K}\sum_{p\in\mathcal{P}} \mathbf{1}\left[z^{(i)}_{i,p} = 1 \wedge z^{(K)}_{i,p} = 0\right]}{\sum_{i=1}^{K}\sum_{p\in\mathcal{P}} z^{(i)}_{i,p}}. \tag{16}$$

Final Success averages the final forgetting indicators over all $2K$ cases. The Reappearance Rate denominator contains the cases successfully forgotten at their immediate checkpoints, and its numerator counts the same cases that fail the criterion under checkpoint θ_K.

CLIP-IQA. We measure generation quality using CLIP-IQA. We use the TorchMetrics `CLIPImageQualityAssessment` implementation with the `clip_iqa` model, the `quality` prompt, and data range 1.0. The score at each stage averages the 32 images generated for the current target, comprising 16 images from each prompt type. Previous-target samples generated at the reported prefix checkpoints are used only for memorization evaluation and do not enter the CLIP-IQA

average. We apply the same KMeans and CLIP-IQA evaluation pipeline to all methods, including Prompt-Free.

Compared methods and optimization details. We compare LASTING with SISS, EraseDiff, ReTrack, Forward KL-Constrained Unlearning, and Prompt-Free Instance Unlearning. Table 16 reports the configuration used for each sequential run.

Table 16: Optimization settings for the Stable Diffusion experiments. Updates denote optimizer updates for each deletion target.

Method	Optimizer	Learning rate	Batch	Accum.	Updates per target	Method-specific setting
SISS	AdamW	1×10^{-5}	1	16	35	$\lambda = 0.5$, `scaling_norm`$= 750$
EraseDiff	AdamW	1×10^{-5}	1	16	35	$\eta = 1 \times 10^{-2}$
ReTrack	AdamW	1×10^{-5}	1	16	30	$k = 10$, $\lambda = 0.5$, latent ℓ_2 distance
FU	Adam	1×10^{-5}	1	8	40	$\rho = 1.25$, dual learning rate 5×10^{-2}
Prompt-Free	AdamW	1×10^{-5}	1	4	60	$\beta = 5 \times 10^{-5}$, global forget-gradient projection
LASTING	AdamW	1×10^{-5}	1	16	35	$\epsilon_{\text{nbr}} = 0.05$, $\eta_{\text{in}} = 0.05$, $J = 1$

Common implementation. All methods optimize the U-Net while keeping the text encoder and VAE fixed. We use a constant learning-rate schedule without warmup and clip the maximum gradient norm at 1.0. The AdamW runs use $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.01, and Adam epsilon 10^{-8}. All methods follow the same target order and use the same stage-specific memorized and non-memorized partitions.

Baseline optimization. EraseDiff matches the retain prediction to diffusion noise and the forget prediction to uniform random noise. ReTrack constructs target-specific neighbors from the non-memorized partition using VAE-latent ℓ_2 distance and combines its unlearning and retain losses using the weight reported in Table 16. Prompt-Free Instance Unlearning projects the forget gradient when its global inner product with the retain gradient is negative and then combines it with the retain gradient.

FU calibration. We calibrate FU on the first deletion target over $\rho \in \{1.25, 1.5, 2.0\}$, primal learning rates $\{5 \times 10^{-6}, 1 \times 10^{-5}\}$, dual learning rates $\{1 \times 10^{-2}, 5 \times 10^{-2}\}$, and optimizer-update budgets $\{20, 40, 60\}$. Eligible candidates achieve a forget-ratio attainment rate of at least 0.9 while limiting the relative retain-loss increase to at most 0.1. Among these candidates, we select the configuration with the lowest mean SSCD and use it for the full sequence. FU applies ρ to the θ_0-normalized forget-loss ratio. The dual multiplier is initialized to zero and updated using an EMA coefficient of 0.9.

LASTING *configuration.* LASTING retains the SISS outer objective and optimizer. Its inner search starts from zero perturbation and minimizes the raw latent-diffusion noise-prediction loss through projected sign-gradient descent. The inner search and outer deletion update reuse the same diffusion timestep and Gaussian noise. We detach the selected latent neighbor before evaluating the SISS deletion objective.

Computational environment. We run each experiment on a single NVIDIA RTX 5090 GPU with 32 GB of memory. A complete 30-stage training sequence requires approximately five to six hours per method.

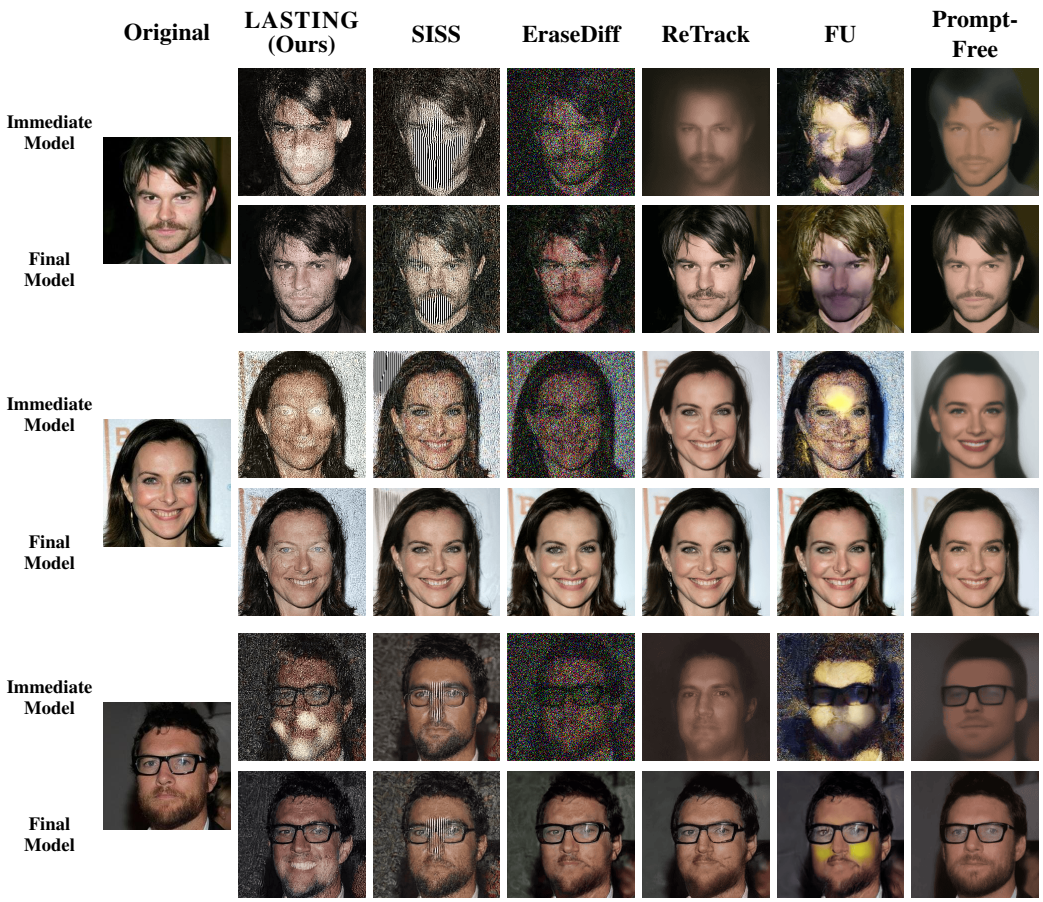


Figure 4: Additional qualitative results on CelebA-HQ. For each deletion target, the immediate model shows the recovery output immediately after that target is unlearned. The final model shows the recovery output after all subsequent deletion requests have been processed.

D Additional Qualitative Results

D.1 CelebA-HQ

Figure 4 presents six additional deletion targets from the CelebA-HQ sequence. For each target, comparing the immediate and final reconstructions shows how its visual resemblance changes as subsequent deletion requests are processed. These examples provide target-level context for the aggregate persistence results reported in the main paper.

D.2 Stable Diffusion

Figure 5 presents four additional memorized prompts and their reference images. The immediate and final generation composites allow visual inspection of whether reference-specific content remains suppressed after subsequent deletion requests. Showing four generations per checkpoint also makes variation within each prompt visible.

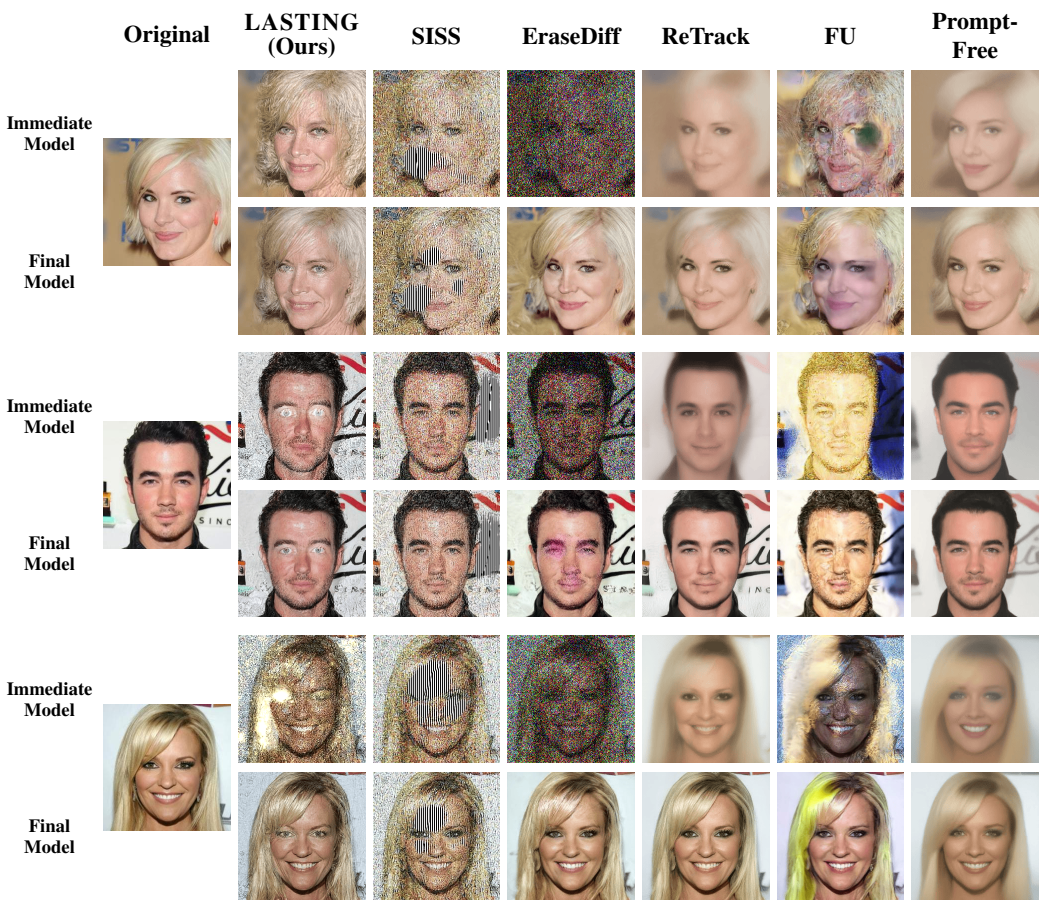


Figure 4: Additional qualitative results on CelebA-HQ (continued).



Figure 5: Additional qualitative results on Stable Diffusion. Each prompt is divided into two method groups, with the reference image repeated in each group. Each method image is a 2×2 composite of four generations. Immediate and final outputs use the same prompt and four random seeds across methods.

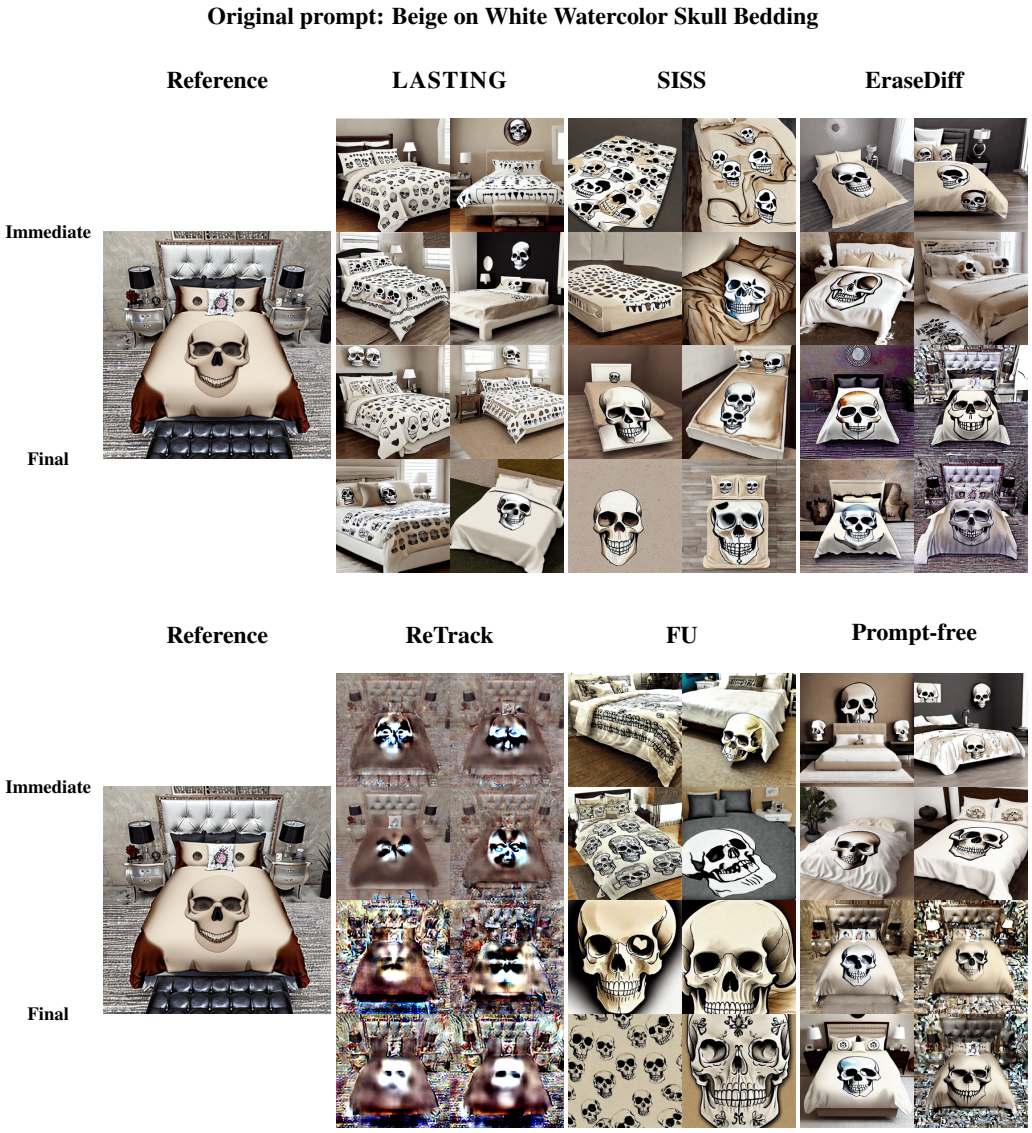


Figure 5: Additional qualitative results on Stable Diffusion (continued).



Figure 5: Additional qualitative results on Stable Diffusion (continued).

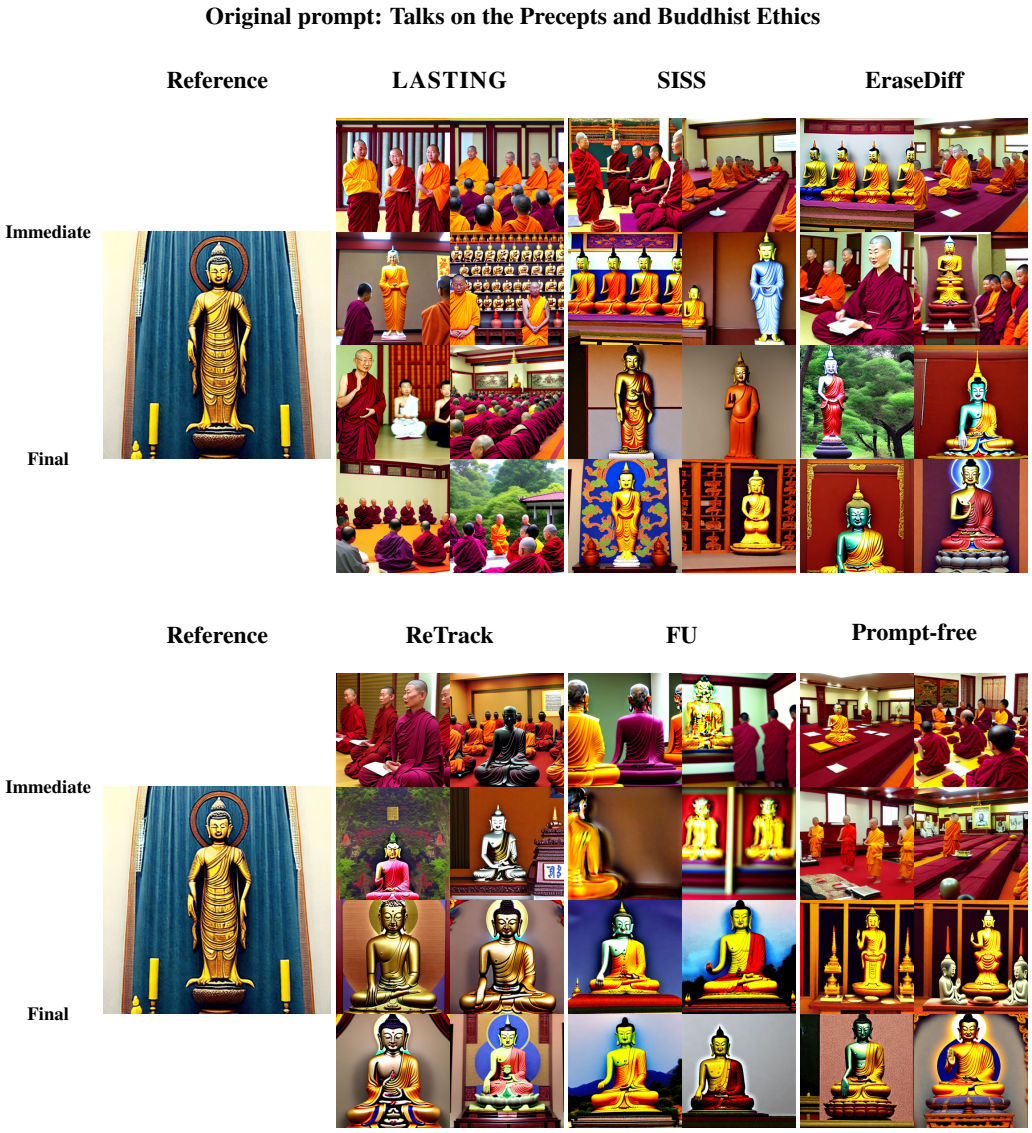


Figure 5: Additional qualitative results on Stable Diffusion (continued).