FLOWFORGET: DATA UNLEARNING IN FLOW MATCHING MODELS VIA ON-POLICY DISTILLATION AND PREFERENCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow Matching has emerged as an effective paradigm for modern generative modeling and often relies heavily on large-scale training data. Some training samples may later need to be removed for privacy, safety, or copyright reasons. In this paper, we focus on flow matching data unlearning, which aims to remove the influence of specified training samples while preserving overall generative quality. This task faces two key challenges: preserving model utility during forgetting and constructing effective sample-level forgetting signals. Existing methods typically preserve utility with fixed reference targets, which may poorly reflect the model's evolving generation distribution. Moreover, flow matching unlearning largely focuses on concept-level erasure, with limited support for individual-sample removal. To address these challenges, we propose **FlowForget**, an on-policy distillation and preference optimization framework for flow matching data unlearning. **FlowForget** transforms the deterministic flow ordinary differential equation into a stochastic process that preserves its marginal distributions, yielding non-degenerate transition policies for policy matching during generation. It then performs teacher-guided on-policy distillation along the model's current trajectories to preserve generative behavior using retained data. Furthermore, **FlowForget** evaluates sampled generations with a joint reward and applies group relative policy optimization over stochastic trajectories to obtain adaptive forgetting signals. Extensive experiments on real-world datasets demonstrate that **FlowForget** achieves effective data unlearning while maintaining generative quality.

1 INTRODUCTION

Flow Matching has recently demonstrated strong generative modeling capabilities by learning a time-dependent velocity field that continuously transports samples from a simple prior distribution to a complex data distribution (Chen & Lipman, 2024; Liu et al., 2026; Kornilov et al., 2024). However, such models typically rely on large-scale training data, and even after training is completed (Rao & Moyer, 2026; Wu et al., 2025b), some training samples may still need to be removed due to privacy or safety concerns (Kusumba et al., 2026; Simone et al., 2025). Meanwhile, regulations such as the General Data Protection Regulation (GDPR), which establishes the *right to be forgotten*, and the California Consumer Privacy Act (CCPA), which provides the right to delete personal information, allow individuals to request the removal of data associated with them (Protection, 2018; Illman & Temple, 2019). A straightforward solution is to remove the specified data and retrain the model on the remaining dataset (Alberti et al., 2025; Zhang et al., 2024). However, retraining from scratch often incurs substantial additional computational and time costs. Therefore, *flow matching unlearning* has emerged to remove the influence of specified training data from a trained model without retraining it from scratch, while preserving its overall generative capability as much as possible.

Existing studies on flow matching unlearning have mainly focused on *concept unlearning*, which aims to remove semantic concepts (e.g., persons, styles, and objects) from the model's generation space. However, practical data deletion requests often target specific training samples involving personal privacy or prohibited content, rather than entire semantic categories. Unlike concept unlearning, *data unlearning* aims to remove the influence of specified training samples while preserving overall generative quality. Concept unlearning can often rely on prompts, classes, or target distribu-

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

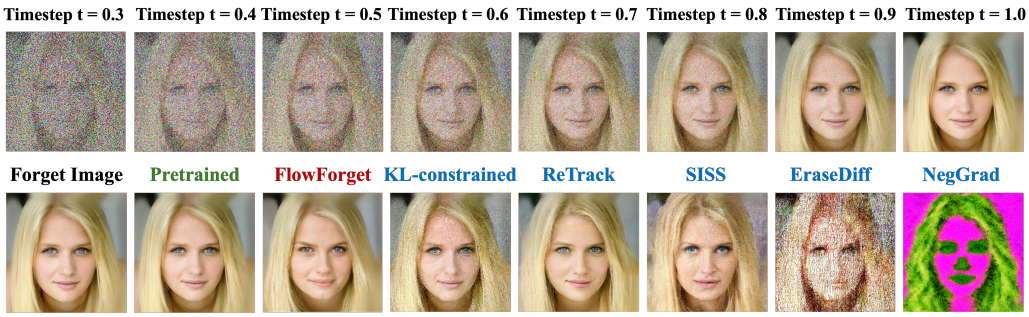


Figure 1: Flow-matching generation process and qualitative comparison of unlearning methods on CelebA-HQ. The first row shows intermediate samples along the flow trajectory, and the second row compares the forget image, pretrained model, and outputs of different unlearning methods.

tions to guide the unlearning process, whereas data unlearning typically lacks such direct guidance for individual training samples. As shown in Fig. 1, existing methods show varying trade-offs between target removal and generation quality, motivating a method that can better balance these two objectives. There are two main challenges for *flow matching data unlearning*: (1) **CH1**: How to preserve overall generative quality during unlearning? (2) **CH2**: How to construct an effective sample-level unlearning signal to sufficiently remove the influence of specified training samples?

However, existing methods remain limited for flow matching data unlearning. On the one hand, they mainly preserve model utility through off-policy supervision (Thakral et al., 2025; Li et al., 2025). SISS (Alberti et al., 2025) combines remaining-data training with forget-data gradient ascent via importance sampling, while EraseDiff (Wu et al., 2025a) couples retained-data denoising with a random-noise forgetting constraint. MACE (Lu et al., 2024) uses concept-focal importance sampling, whereas ReTrack (Shi et al., 2026) redirects forget trajectories toward their k-nearest neighbors and ReCARE (Kim et al., 2026) distills preservation targets from benign co-occurring concepts. These signals may mismatch the current generation distribution, limiting the preservation of original generative behavior. On the other hand, recent methods mainly construct concept-level forgetting signals (Biswas et al., 2026; Li et al., 2026a). EraseFlow (Kusumba et al., 2026) optimizes concept-erasure trajectories with GFlowNets, while CGRU (Vysotskyi et al., 2026) guides denoising with timestep-wise critic rewards. Minimalist Concept Erasure (Zhang et al., 2025) optimizes the distributional distance of final generations, whereas EraseAnything (Gao et al., 2025) employs bi-level LoRA optimization with attention regularization. However, such concept-level guidance cannot directly provide sample-level forgetting signals, limiting its applicability to data unlearning.

To address these challenges, we propose an on-policy distillation and preference optimization framework for flow matching data unlearning, termed **FlowForget**. **FlowForget** includes two main modules, i.e., *trajectory-aligned on-policy distillation* and *unlearning-oriented policy optimization*. In the trajectory-aligned on-policy distillation module, we transform the deterministic flow ODE into a marginal-preserving stochastic process, inducing transition policies for transition-level matching. We then perform on-policy distillation along trajectories visited by the current model, with the teacher model providing supervision at these states, thereby preserving the flow dynamics associated with the retained data and addressing **CH1**. In the unlearning-oriented policy optimization module, We define an unlearning reward based on sample-level forgetting and generation quality, and use group relative policy optimization to optimize relative preferences among sampled trajectories. This provides adaptive sample-level forgetting signals that progressively suppress the influence of the forget samples, thereby addressing **CH2**. By integrating these two modules, **FlowForget** effectively reduces the influence of specified forget samples while preserving generation quality.

We summarize our main contributions as follows: (1) We propose **FlowForget**, a novel framework for flow matching data unlearning that integrates on-policy distillation with preference optimization for effective forgetting and utility preservation. (2) We introduce trajectory-aligned on-policy supervision to preserve generative behavior along evolving trajectories, and construct adaptive sample-level forgetting signals from current generations, with dynamic balancing between forgetting and retention objectives. (3) Extensive experiments on multiple real-world datasets demonstrate that **FlowForget** achieves effective forgetting while well preserving overall generation quality.

2 Related Work

Machine Unlearning. Machine unlearning aims to selectively remove specified information from a trained model while preserving model utility (Li et al., 2024; Patel & Qiu, 2025; Zhong et al., 2025). Existing methods generally fall into exact and approximate unlearning (Zhao et al., 2024): the former aims to recover the model retrained without the forget data, while the latter approximates this behavior at lower computational cost. For exact unlearning, S^3T (Basu Roy Chowdhury et al., 2025) isolates data influence through parameter-efficient training, and some work (Koloskova et al., 2025) provide certified guarantees through noisy model updates. Approximate unlearning updates the trained model by selectively modifying parameters or supervision signals (Kawamura et al., 2026; Xiao et al., 2025). AMUN (Ebrahimpour-Boroojeny et al., 2025) uses adversarial examples to reduce prediction confidence. Meanwhile, LoTUS (Spartalis et al., 2025) smooths predicted probabilities to suppress memorization, and Q-MUL (Tong et al., 2025) replaces original labels with semantically similar alternatives. SalUn (Fan et al., 2024) further uses gradient-based weight saliency to identify and update parameters most relevant to forget samples. Latest, MU-Mis (Cheng et al., 2026) reduce model sensitivity to forget samples to suppress their contribution. However, existing machine unlearning methods are not designed for the continuous generation dynamics of flow matching models, making their direct application to flow matching data unlearning nontrivial.

Concept Unlearning. Concept unlearning aims to suppress specified semantic concepts from generative models while preserving unrelated generation capabilities (George et al., 2025; Moon et al., 2025). It has been extensively studied in text-to-image diffusion models. ESD (Gandikota et al., 2023) erases target concepts through negative guidance, while some work (Kumari et al., 2023) maps them to anchor concepts. SPM (Lyu et al., 2024) learns transferable erasure modules with latent anchoring to limit unintended changes to non-target concepts, whereas AdvUnlearn (Zhang et al., 2024) improves robustness through adversarial training. More recently, GLoCE (Lee et al., 2025) selectively suppresses target concept embeddings and activates erasure only in target-related regions, SuMa (Nguyen et al., 2025) maps target concepts to reference subspaces, and Semantic Surgery (Xiong et al., 2026) removes concepts through semantic subtraction in the text embedding space. Most recently, HiRM (Lee et al., 2026) redirects target representations in the text encoder, SDErasure (Miao et al., 2026) shifts generation trajectories at concept-critical timesteps, and SPEED (Li et al., 2026b) performs scalable erasure through null-space parameter editing. Nonetheless, existing concept unlearning methods typically rely on concept-level information such as prompts or class labels. This makes it difficult to remove the influence of individual training samples.

3 Method

3.1 Problem Formulation

Let $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$ denote the original training dataset of N samples, and let $p_{\mathcal{D}}$ denote its empirical distribution. We consider a pretrained flow matching model $v_{\theta_0}(x_t, t)$ trained on $\mathcal{D}$, with θ_0 denoting its parameters. During unlearning, θ denotes the trainable parameters initialized as $\theta^{(0)} = \theta_0$, and v_θ the corresponding model. Given an initial noise sample $x_0 \sim p_0 = \mathcal{N}(0, I)$, a data sample $x_1 \sim p_{\mathcal{D}}$, and $t \sim \mathcal{U}[0, 1]$, following the Rectified Flow formulation, we construct the linear path $x_t = (1 - t)x_0 + tx_1$, with target velocity $u_t(x_t \mid x_0, x_1) = x_1 - x_0$, which remains constant over t. The flow matching objective for learning this velocity field along the path is:

$$\mathcal{L}_{\mathrm{FM}}(\theta; \mathcal{D}) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_0\sim p_0,\, x_1\sim p_{\mathcal{D}}} \left[\|v_\theta(x_t, t) - (x_1 - x_0)\|_2^2 \right]. \tag{1}$$

At inference, samples are generated by solving the ordinary differential equation (ODE) $\frac{dX_t}{dt} = v_\theta(X_t, t)$ from $X_0 \sim p_0$ over $t \in [0, 1]$, thereby mapping noise samples to the data distribution. Then, given a forget set $\mathcal{D}_f \subset \mathcal{D}$, the retain set is defined as $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$, with $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_f$ and $\mathcal{D}_r \cap \mathcal{D}_f = \varnothing$. Let $v_{\theta_r^\star}$ denote the model retrained from scratch on $\mathcal{D}_r$ by minimizing $\mathcal{L}_{\mathrm{FM}}(\theta; \mathcal{D}_r)$. Let θ_u denote the parameters obtained after unlearning. Ideally, the unlearned model v_{θ_u} should approximate the model retrained on $\mathcal{D}_r$. *Data unlearning* aims to remove the influence of $\mathcal{D}_f$ while preserving the distribution learned from $\mathcal{D}_r$. We define this as the following optimization problem:

$$\min_\theta \; [\mathcal{L}_f(\theta) + \lambda_r \mathcal{L}_r(\theta)], \tag{2}$$

where $\mathcal{L}_f$ and $\mathcal{L}_r$ are the forgetting and retention objectives, respectively, and λ_r controls retention.

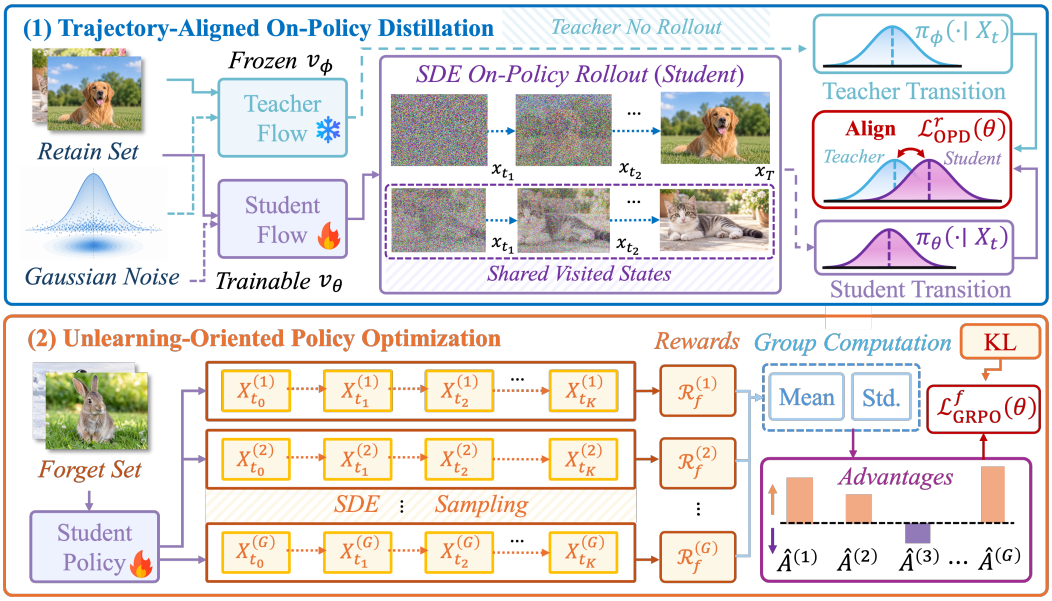


Figure 2: The overall framework of our proposed FlowForget.

In the following, we focus on *data unlearning* for flow matching models and propose **FlowForget**, which consists of *trajectory-aligned on-policy distillation* and *unlearning-oriented policy optimization*. **FlowForget** preserves generative behavior through on-policy distillation while using reward-guided policy optimization to reduce the influence of forget samples. It aims to approximate a model retrained on $\mathcal{D}_r$ without full retraining. The overall framework is shown in Fig. 2, with the main notation summarized in Appendix A and algorithmic details provided in Appendix B.

3.2 Trajectory-Aligned On-Policy Distillation

In this section, we address retention by aligning v_θ with v_ϕ on retain-side trajectories anchored by $\mathcal{D}_r$. A natural strategy is to distill v_ϕ into v_θ on predefined states constructed from $\mathcal{D}_r$. However, these states may differ from those visited by the evolving model, causing rollout mismatch and accumulated local prediction errors during generation. Therefore, we propose *trajectory-aligned on-policy distillation*. Given $x_r \sim \mathcal{D}_r$, we construct a retain-anchored state $X^r_{t_0} = (1-t_0)\epsilon + t_0 x_r$, where $\epsilon \sim \mathcal{N}(0, I)$, and roll out the current student from $X^r_{t_0}$ to obtain the states it actually visits. This induces the retain-side on-policy distribution $X_t \sim \rho^r_{\theta,t}$, where the frozen teacher model v_ϕ provides dense velocity targets $v_\phi(X_t, t)$ at the student-visited states. Thus, the distillation signal remains anchored to the retain data while adapting to the student's evolving trajectory distribution. To cast distillation as policy matching, we introduce stochastic local transitions, since the deterministic flow ODE $dX_t = v_\theta(X_t, t)dt$ yields degenerate conditional transitions and cannot support student–teacher policy matching. We therefore associate the flow dynamics with an SDE whose transition kernel defines the non-degenerate local policies required for transition-level matching.

Proposition 3.1 (Marginal-Preserving Stochastic Flow). *For the linear flow-matching path $X_t = (1-t)X_0 + tX_1$, let p_t denote its time-marginal distribution. For a velocity field satisfying the flow-matching optimum $v_\theta(x,t) = \mathbb{E}[X_1 - X_0 \mid X_t = x]$, the flow ODE $dX_t = v_\theta(X_t, t)dt$ can be associated with the following marginal-preserving stochastic counterpart for $t \in (0,1)$:*

$$dX_t = \left[v_\theta(X_t, t) - \frac{\sigma_t^2}{2(1-t)} \left(X_t - t v_\theta(X_t, t) \right) \right] dt + \sigma_t dW_t, \tag{3}$$

where W_t is a standard Wiener process and $\sigma_t = \kappa\sqrt{(1-t)/t}$ controls stochasticity. Under this condition, the SDE and flow ODE share the same time-marginal distribution p_t for $t \in (0,1)$.

The proof of Proposition 3.1 is provided in Appendix C.1. In retain-side distillation, we initialize the process from $X^r_{t_0}$ and roll out the student using the transition kernel in Eq. (3). The injected stochasticity turns deterministic updates into non-degenerate transition distributions, enabling policy

matching between the student and teacher models. After Euler–Maruyama discretization, the transition mean depends on v_θ, yielding the Gaussian transition policy below.

Proposition 3.2 (Gaussian Transition Policy)**.** *Given the Euler–Maruyama discretization of Eq. (3) with $t \in (0,1)$, $\Delta t > 0$, and $t + \Delta t \leq 1$, the transition $X_t \to X_{t+\Delta t}$ is isotropic Gaussian:*

$$\begin{aligned} \pi_\theta(X_{t+\Delta t} \mid X_t) &= \mathcal{N}\big(\mu_\theta(X_t,t), \sigma_t^2 \Delta t I\big)\,, \\ \mu_\theta(X_t,t) &= X_t + \left[v_\theta(X_t,t) - \frac{\sigma_t^2}{2(1-t)}\big(X_t - t v_\theta(X_t,t)\big)\right]\Delta t. \end{aligned} \tag{4}$$

The proof of Proposition 3.2 is provided in Appendix C.2. We cast retain-side distillation as local policy matching between $\pi_\theta(\cdot \mid X_t)$ and $\pi_\phi(\cdot \mid X_t)$ for $X_t \sim \rho^r_{\theta,t}$ sampled along the trajectories. This aligns the local transition behavior of v_θ with that of v_ϕ along the visited states. Since π_θ and π_ϕ share covariance $\sigma_t^2 \Delta t I$, their KL divergence reduces to a weighted velocity-space discrepancy.

Proposition 3.3 (Velocity-Space KL Reduction)**.** *The Gaussian transition policies induced by v_θ and v_ϕ at X_t share covariance $\sigma_t^2 \Delta t I$. Their transition-level KL divergence is given by*

$$D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid X_t) \,\|\, \pi_\phi(\cdot \mid X_t)\right) = \frac{\Delta t}{2}\left(\frac{1}{\sigma_t} + \frac{\sigma_t t}{2(1-t)}\right)^2 \|v_\theta(X_t,t) - v_\phi(X_t,t)\|_2^2\,. \tag{5}$$

The proof of Proposition 3.3 is provided in Appendix C.3. This result establishes a direct connection between transition-level policy matching and velocity-field alignment at on-policy retain states, allowing the retention objective to be optimized in the velocity space. We therefore define

$$\mathcal{L}^r_{\mathrm{OPD}}(\theta) = \mathbb{E}_{t,\, X_t \sim \rho^r_{\theta,t}}\left[D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid X_t) \,\|\, \pi_\phi(\cdot \mid X_t)\right)\right]. \tag{6}$$

We minimize $\mathcal{L}^r_{\mathrm{OPD}}$ to match the student transition policy with the teacher policy along visited retain trajectories. This transition-level objective limits deviations in local dynamics induced by forget-side updates, thereby preserving retain-side flow dynamics throughout unlearning.

3.3 Unlearning-Oriented Policy Optimization

To suppress the influence of $\mathcal{D}_f$, we formulate forget-side unlearning as a reward-guided trajectory optimization problem, where the forgetting behavior of trajectories sampled from the current policy is directly evaluated by an unlearning reward as the model evolves during unlearning. Based on this formulation, we introduce *unlearning-oriented policy optimization* to redirect the generation trajectories of v_θ away from forget-related behaviors through online reinforcement learning over $\pi_\theta(\cdot \mid X_t)$. Specifically, for each $x_f \sim p_{\mathcal{D}_f}$, we sample a group of trajectories $\{\tau_i\}_{i=1}^G$ from the current policy at each iteration, producing final generated outputs $\{\hat{x}_i\}_{i=1}^G$, and evaluate each trajectory with the unlearning reward $r_i = \mathcal{R}_f(\hat{x}_i, x_f)$. GRPO then directly derives group-relative advantages from these rewards to progressively steer π_θ away from memorized behaviors. This group-relative formulation enables online adaptation to the evolving policy without an additional value model and supports non-differentiable unlearning signals.

Unlearning-oriented reward. To realize these preferences, for each final generated output $\hat{x}_i$ and its corresponding forget sample x_f, we define the unlearning reward $r_i = \mathcal{R}_f(\hat{x}_i, x_f)$ as

$$r_i = a\,\mathcal{F}(\hat{x}_i, x_f) + (1-a)\frac{1}{M}\sum_{m=1}^{M}\mathcal{U}_m(\hat{x}_i), \tag{7}$$

where $\mathcal{F}(\hat{x}_i, x_f)$ measures the degree of forgetting with respect to x_f by rewarding outputs that deviate from the memorized content associated with x_f, while $\{\mathcal{U}_m(\hat{x}_i)\}_{m=1}^M$ denote M complementary rewards that preserve different aspects of generation quality and discourage undesirable degradation during unlearning. The coefficient $a \in [0,1]$ controls the relative contribution of forgetting and quality preservation to r_i. Consequently, a larger r_i favors $\hat{x}_i$ with stronger forgetting of x_f while maintaining generation quality. Since memorization and generation quality are evaluated differently across tasks, the task-specific reward settings for $\mathcal{F}$ and $\{\mathcal{U}_m\}_{m=1}^M$ are provided in Appendix E.

Group-relative policy optimization. Using the reward defined above, we optimize the policy according to the relative preferences among sampled trajectories with GRPO. To enable stochastic trajectory sampling during policy optimization, we use the flow-based SDE in Eq. (3), whose

discretization induces the Gaussian transition policy in Proposition 3.2. For each forget sample $x_f \sim p_{\mathcal{D}_f}$, the behavior policy $\pi_{\theta_{\text{old}}}$ samples a group of G stochastic trajectories $\{\tau_i\}_{i=1}^{G}$, where $\tau_i = (X_{t_0}^{(i)}, X_{t_1}^{(i)}, \ldots, X_{t_K}^{(i)})$ with $0 < t_0 < t_1 < \cdots < t_K = 1$, and the final generated output is $\hat{x}_i = X_{t_K}^{(i)}$. Here, θ_{old} denotes the policy parameters before the GRPO update and remains fixed for the sampled trajectories. Then, we compute r_i and normalize the rewards within the group to obtain

$$\widehat{A}_i = \frac{r_i - \bar{r}}{s_r + \epsilon}, \quad \bar{r} = \frac{1}{G}\sum_{j=1}^{G} r_j, \quad s_r = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \bar{r})^2}. \tag{8}$$

where $\epsilon > 0$ is a small constant for numerical stability. Since all trajectories within a group share the same normalization statistics, i.e., the same group mean and standard deviation, their reward ordering is preserved under this transformation, such that $r_i > r_j \Leftrightarrow \widehat{A}_i > \widehat{A}_j$ for any $i, j \in \{1, \ldots, G\}$. Thus, $\widehat{A}_i$ represents the relative preference of τ_i with respect to the other trajectories within the same group. The same trajectory-level advantage is used for all transition-wise policy ratios $\rho_{k,i}(\theta), k = 0, \ldots, K-1$. Given the Gaussian transition of π_θ established in Proposition 3.2, the transition-wise importance ratio between the current policy and the behavior policy is tractable:

$$\rho_{k,i}(\theta) = \frac{\pi_\theta\left(X_{t_{k+1}}^{(i)} \mid X_{t_k}^{(i)}\right)}{\pi_{\theta_{\text{old}}}\left(X_{t_{k+1}}^{(i)} \mid X_{t_k}^{(i)}\right)}. \tag{9}$$

We regularize π_θ toward the fixed reference policy π_{θ_0} induced by v_{θ_0} to limit excessive deviation during reward-driven optimization throughout the unlearning process. For each transition, we define

$$D_{k,i}^{\text{KL}}(\theta) = D_{\text{KL}}\left(\pi_\theta(\cdot \mid X_{t_k}^{(i)}) \| \pi_{\theta_0}(\cdot \mid X_{t_k}^{(i)})\right). \tag{10}$$

We then optimize π_θ using the following KL-regularized clipped group-relative surrogate objective:

$$\mathcal{J}_f(\theta) = \mathbb{E}_{\substack{x_f \sim p_{\mathcal{D}_f} \\ \{\tau_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}}} \left[\frac{1}{GK} \sum_{i=1}^{G} \sum_{k=0}^{K-1} \left(\min\left(\rho_{k,i}(\theta)\widehat{A}_i, \text{clip}\left(\rho_{k,i}(\theta), 1-\varepsilon, 1+\varepsilon\right)\widehat{A}_i\right) - \beta D_{k,i}^{\text{KL}}(\theta) \right) \right]. \tag{11}$$

Here, ε specifies the clipping range, and β controls the KL regularization strength. Since $\mathcal{J}_f(\theta)$ is formulated as a maximization objective, we define the corresponding minimization objective as $\mathcal{L}_{\text{GRPO}}^{f}(\theta) = -\mathcal{J}_f(\theta)$, which is combined with the retain-side objective below in joint optimization.

Joint optimization. After deriving the forget-side objective $\mathcal{L}_{\text{GRPO}}^{f}(\theta)$ and the retain-side objective $\mathcal{L}_{\text{OPD}}^{r}$, we jointly optimize the two objectives over θ. Existing methods typically combine $\mathcal{L}_{\text{GRPO}}^{f}(\theta)$ and $\mathcal{L}_{\text{OPD}}^{r}$ with a fixed coefficient. However, such static weighting cannot account for their evolving optimization scales. As their gradient magnitudes change over training, the contributions of $\mathcal{L}_{\text{GRPO}}^{f}(\theta)$ and $\mathcal{L}_{\text{OPD}}^{r}$ to the update of θ may become imbalanced, causing one objective to dominate the other. We therefore balance their contributions according to the current gradient magnitudes. At optimization step k, we compute $g_f^{(k)} = \nabla_\theta \mathcal{L}_{\text{GRPO}}^{f,(k)}$ and $g_r^{(k)} = \nabla_\theta \mathcal{L}_{\text{OPD}}^{r,(k)}$. We then control the relative contribution of the retain-side update by targeting $\|\lambda_r^{(k)} g_r^{(k)}\|_2 \approx \rho \|g_f^{(k)}\|_2$, where $\rho > 0$ specifies the desired retention strength relative to the forgetting update. Thus, we set

$$\lambda_r^{(k)} = \rho \frac{\|g_f^{(k)}\|_2}{\|g_r^{(k)}\|_2 + \delta}, \tag{12}$$

where δ ensures numerical stability. Thus, $\lambda_r^{(k)}$ increases when $\|g_f^{(k)}\|_2$ grows relative to $\|g_r^{(k)}\|_2$, and decreases when $\|g_r^{(k)}\|_2$ becomes larger. We stop gradients through $\lambda_r^{(k)}$ when updating θ. This adjustment compensates for the scale asymmetry between $\mathcal{L}_{\text{GRPO}}^{f}$ and $\mathcal{L}_{\text{OPD}}^{r}$ without altering their optimization targets, keeping the retain-side update commensurate with the evolving forget-side update and avoiding persistent dominance by either objective. The overall objective is

$$\mathcal{L}_{\text{FlowForget}}^{(k)}(\theta) = \mathcal{L}_{\text{GRPO}}^{f,(k)}(\theta) + \lambda_r^{(k)} \mathcal{L}_{\text{OPD}}^{r,(k)}(\theta). \tag{13}$$

The objective combines reward-guided forgetting and retain-side on-policy distillation within a unified parameter update through gradient combination. In summary, **FlowForget** suppresses the influence of $\mathcal{D}_f$ while preserving the flow dynamics associated with $\mathcal{D}_r$ during optimization, enabling targeted and efficient data unlearning in flow matching models without requiring full retraining.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We conduct experiments on commonly used real-world datasets, i.e., *MNIST T-Shirt*, *CelebA-HQ*, and *CIFAR-10*. MNIST T-Shirt is constructed from MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017), where MNIST provides the retain distribution and a T-shirt image from Fashion-MNIST is introduced as the designated forget sample. CelebA-HQ (Karras et al., 2017) and CIFAR-10 (Krizhevsky et al., 2009) provide more complex face and natural object images, respectively. Detailed dataset statistics and construction protocols are provided in Appendix D.1.

Compared Methods. We compare our proposed **FlowForget** with a set of representative unlearning baselines, including **Pretrained**, **KL-constrained**(Khalafi et al., 2026), **ReTrack** (Shi et al., 2026), **SISS** (Alberti et al., 2025), **EraseDiff** (Wu et al., 2025a), and **NegGrad** (Golatkar et al., 2020). The details of the unlearning baselines are provided in Appendix D.2.

Evaluation Metrics. We evaluate **FlowForget** in terms of both *unlearning effectiveness* and *generation quality*. (1) For *MNIST T-Shirt*, we use negative log likelihood (NLL) (Song et al., 2020) and T-shirt generation frequency p (Alberti et al., 2025) to evaluate unlearning effectiveness. Here, p is defined as the proportion of target T-shirt samples among 30,720 generated images. Following SISS (Alberti et al., 2025), a generated sample is identified as the target T-shirt when its pixel-space ℓ_2 distance to the designated T-shirt image is smaller than 10. Inception Score (IS) (Salimans et al., 2016) is used to assess generation quality. (2) For *CelebA-HQ* and *CIFAR-10*, we use NLL and SSCD (Pizzi et al., 2022) to evaluate unlearning effectiveness. SSCD measures the similarity between generated samples and forget samples, while Frechet Inception Distance (FID) (Heusel et al., 2017) is used to assess generation quality. Higher NLL and lower SSCD or p indicate stronger forgetting, while higher IS and lower FID indicate better generation quality.

Implementation Details. We summarize the implementation details as follows. (1) For *MNIST T-Shirt*, both the teacher and student use UNet architectures to parameterize the flow matching velocity field. We optimize the student using AdamW with a learning rate of 5×10^{-5}. At each rollout iteration, we sample 32 trajectories organized into four groups with group size $G = 8$ and use $K = 8$ stochastic rollout steps. We set the SDE noise level to $\kappa = 0.1$. For policy optimization, we set the clipping parameter to $\epsilon = 10^{-4}$ and the KL regularization coefficient to $\beta = 0.02$. We set the relative retention strength to $\rho = 1.0$. (2) For *CelebA-HQ*, both the teacher and student use NCSN++ architectures to parameterize the Flow Matching velocity field. We optimize the student using AdamW with a learning rate of 5×10^{-6}. At each rollout iteration, we sample 32 trajectories organized into four groups with group size $G = 8$ and use $K = 10$ stochastic rollout steps. We set the SDE noise level to $\kappa = 0.7$. For policy optimization, we set the clipping parameter to $\epsilon = 10^{-4}$ and the KL regularization coefficient to $\beta = 0.02$. We set the relative retention strength to $\rho = 1.0$. (3) For *CIFAR-10*, both the teacher and student use DDPM++-style UNet architectures to parameterize the Flow Matching velocity field. We optimize the student using AdamW with a learning rate of 2×10^{-5}. At each rollout iteration, we sample 32 trajectories organized into four groups with group size $G = 8$ and use $K = 10$ stochastic rollout steps. We set the SDE noise level to $\kappa = 0.1$. For policy optimization, we set the clipping parameter to $\epsilon = 10^{-4}$ and the KL regularization coefficient to $\beta = 0.02$. We set the relative retention strength to $\rho = 1.0$. Additional implementation details (e.g., pretrained model and environment configurations) are provided in Appendix D.3.

4.2 PERFORMANCE COMPARISON ON MNIST T-SHIRT

Unlearning Effectiveness. The comparison results on *MNIST T-Shirt* are reported in Table 1 and Fig. 3. We observe that: (1) The pretrained model generates recognizable T-shirt, indicating that the designated sample influences its generation behavior. (2) The baselines show trade-offs between forgetting effectiveness and generation quality. ReTrack and SISS suppress the target sample while largely preserving handwritten digits, whereas EraseDiff still generates samples resembling the target T-shirt, indicating incomplete forgetting. In contrast, KL-constrained and NegGrad degrade the generation distribution and produce low-quality samples, suggesting that insufficiently constrained forgetting updates may impair retained generation. (3) **FlowForget** suppresses the designated T-shirt sample while preserving diverse handwritten digits. This suggests that on-policy optimization steers trajectories away from the memorized sample while limiting changes to the generation distribution.

Table 1: Experimental results on *MNIST T-Shirt*. Results are reported as mean ± standard deviation.

Method	Unlearning		Quality
	NLL↑	p (%)↓	IS↑
Pretrained	0.934 ± 0.061	1.6862 ± 0.0743	9.648 ± 0.026
NegGrad	15.326 ± 1.117	0.0000 ± 0.0000	1.713 ± 0.297
EraseDiff	10.742 ± 0.583	0.0003 ± 0.0010	8.872 ± 0.341
SISS	8.214 ± 0.437	0.0000 ± 0.0000	9.364 ± 0.217
ReTrack	8.361 ± 0.074	0.0000 ± 0.0000	9.487 ± 0.094
KL-constrained	11.486 ± 0.926	0.0000 ± 0.0000	3.926 ± 0.463
FlowForget	8.793 ± 0.092	0.0000 ± 0.0000	9.529 ± 0.071

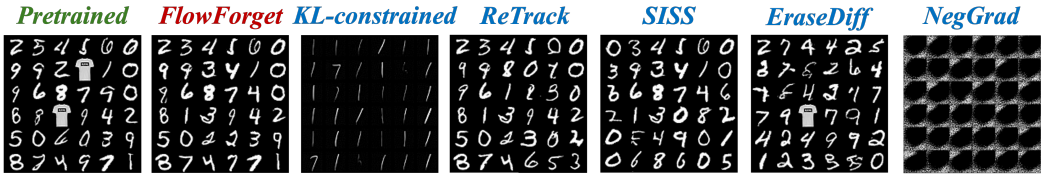


Figure 3: Qualitative comparison of different unlearning methods on contaminated MNIST T-Shirt.

Generation Quality. We evaluate generation quality after unlearning using Table 1 and Fig. 3. We observe that: (1) The pretrained model produces clear and diverse handwritten digits. ReTrack and SISS largely preserve this quality, highlighting the importance of constraining changes to the retained distribution. (2) KL-constrained, EraseDiff, and NegGrad show varying quality degradation. KL-constrained produces low-contrast, less recognizable digits, while NegGrad suffers severe generation collapse, indicating that insufficient retention may distort the learned distribution. (3) **FlowForget** maintains generation behavior close to the pretrained model, producing recognizable digits with diverse styles. We attribute this to trajectory-aligned on-policy distillation, which continuously aligns the student with the frozen teacher along its current trajectories.

4.3 Performance Comparison on CelebA-HQ

Unlearning Effectiveness. The results on *CelebA-HQ* are reported in Table 2 and Fig. 1. We make the following observations: (1) The pretrained model reconstructs a face highly similar to the designated forget image, indicating that information from the sample has been retained in the model. (2) ReTrack and SISS alter the reconstructed face while preserving plausible facial structure, whereas KL-constrained, EraseDiff, and NegGrad cause larger deviations with artifacts or distributional distortion. This indicates that deviation from the forget sample alone does not ensure effective unlearning, as excessive updates may damage the generative distribution. (3) **FlowForget** generates a face that differs from the designated sample while remaining on the natural face distribution. We attribute this to on-policy optimization, which progressively redirects trajectories away from the forget sample and provides more targeted forgetting signals throughout the entire flow process.

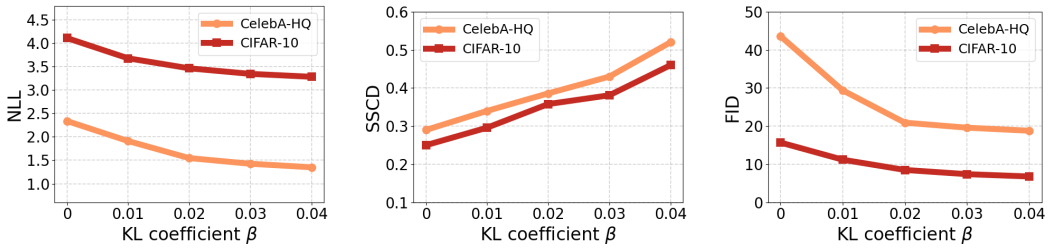
Generation Quality. We evaluate the generation quality after unlearning based on Table 2 and Fig. 1 and make the following observations. (1) ReTrack and SISS preserve realistic facial structures after unlearning. This behavior is consistent with their explicit mechanisms for restricting destructive updates to the original generative distribution. (2) In contrast, KL-constrained, EraseDiff, and NegGrad show varying degrees of visual degradation. KL-constrained and EraseDiff introduce noise and artifacts, while NegGrad causes generation collapse, suggesting that poorly constrained updates can perturb the shared flow field and degrade the data distribution. (3) **FlowForget** maintains realistic face generation while changing the output associated with the forget sample. This is achieved by steering the generation away from the forget sample while preserving the learned flow distribution.

4.4 Performance Comparison on CIFAR-10

Unlearning Effectiveness. As shown in Table 2, ReTrack and SISS achieve effective forgetting while largely preserving the original generation behavior during the unlearning process, whereas

Table 2: Experimental comparison on *CelebA-HQ* and *CIFAR-10*.

Method	CelebA-HQ			CIFAR-10		
	NLL↑	SSCD↓	FID↓	NLL↑	SSCD↓	FID↓
Pretrained	1.291 ± 0.128	0.881 ± 0.023	17.984 ± 0.436	3.224 ± 0.361	0.543 ± 0.059	5.273 ± 0.214
NegGrad	6.872 ± 1.143	0.129 ± 0.042	427.316 ± 48.271	4.392 ± 0.368	0.217 ± 0.064	83.517 ± 31.946
EraseDiff	2.647 ± 0.286	0.337 ± 0.047	110.853 ± 17.583	3.579 ± 0.312	0.469 ± 0.082	27.864 ± 7.953
SISS	1.431 ± 0.245	0.512 ± 0.096	22.387 ± 1.731	3.213 ± 0.356	0.452 ± 0.053	9.176 ± 0.705
ReTrack	1.486 ± 0.227	0.463 ± 0.073	20.756 ± 1.362	3.348 ± 0.291	0.428 ± 0.071	8.694 ± 0.527
KL-constrained	2.194 ± 0.361	0.304 ± 0.068	58.127 ± 8.214	3.796 ± 0.437	0.336 ± 0.076	19.284 ± 4.392
FlowForget	1.547 ± 0.236	0.386 ± 0.091	20.918 ± 1.642	3.467 ± 0.325	0.358 ± 0.046	8.512 ± 0.488



(a) NLL (b) SSCD (c) FID

Figure 4: Sensitivity analysis of the KL regularization coefficient β on *CelebA-HQ* and *CIFAR-10*.

EraseDiff and NegGrad tend to introduce larger changes to the learned distribution. In comparison, **FlowForget** effectively reduces the influence of the forget samples by steering the generation away from them while explicitly constraining undesired changes to the learned flow distribution.

Generation Quality. From Table 2, we observe that ReTrack and SISS maintain stable overall generation quality after unlearning, while KL-constrained, EraseDiff, and NegGrad exhibit different degrees of visible degradation in the generated samples. **FlowForget** effectively reduces the influence of the forget samples while maintaining generation quality close to that of the pretrained model, demonstrating effective data unlearning without degrading the learned generative distribution.

4.5 Ablation and Parameter Analysis

Ablation Study. The detailed ablation results are provided in Appendix F.3. We evaluate the contributions of **Trajectory-Aligned On-Policy Distillation (TA-OPD)** and **Unlearning-Oriented Policy Optimization (UOPO)** by removing each component. Without TA-OPD, forgetting remains effective but generation quality degrades. Without UOPO, generation quality is preserved, but effective forgetting is lost. The full **FlowForget** balances forgetting and generation quality.

Effect of KL Regularization. We evaluate sensitivity to the KL regularization coefficient β by varying $\beta \in \{0, 0.01, 0.02, 0.03, 0.04\}$ and reporting the results in Fig. 4. A small β provides insufficient regularization, allowing larger policy deviations and degrading generation quality. Conversely, a large β overly restricts the policy update, limiting trajectory shifts away from the forget samples and weakening forgetting effectiveness. Therefore, we set $\beta = 0.02$ for **FlowForget**.

5 Conclusion

In this paper, we propose **FlowForget**, an on-policy distillation and preference optimization framework for flow matching data unlearning. **FlowForget** first introduces **trajectory-aligned on-policy distillation** on retained samples, allowing the student to follow its evolving trajectories while preserving the learned flow field under the guidance of a frozen teacher. It then employs **unlearning-oriented policy optimization** on forget samples, using trajectory-level feedback to progressively steer generation away from the designated data. To balance these two objectives, we introduce an adaptive weighting strategy that adjusts the retention strength according to the optimization dynamics of forgetting and retention. Extensive experiments on real-world datasets demonstrate that **FlowForget** achieves strong forgetting effectiveness while maintaining generation quality.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and manuscript polishing, including improving grammar, clarity, fluency, and academic expression. Generative AI was not used to develop the research ideas or methodology, conduct experiments, analyze results, or generate citations. All AI-assisted content was reviewed and revised by the authors for technical accuracy and consistency with the work. We take full responsibility for the final content of this paper.

ETHICS STATEMENT

This work focuses on improving the privacy and safety of generative models through flow matching data unlearning. Our experiments use publicly available research datasets, including MNIST T-Shirt, CelebA-HQ, and CIFAR-10, solely to evaluate the effectiveness of removing designated training samples from generative models. In particular, CelebA-HQ contains facial images, which may involve privacy considerations; we use the dataset only for research evaluation and do not attempt to identify or infer sensitive attributes of individuals. All datasets and model resources are used in accordance with their intended research purposes and applicable terms. No new personal data are collected, and no human subjects are involved in any stage of the experiments.

REFERENCES

Silas Alberti, Kenan Hasanaliyev, Manav Shah, and Stefano Ermon. Data unlearning in diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 3084–3100, 2025.

Somnath Basu Roy Chowdhury, Krzysztof Choromanski, Arijit Sehanobish, Kumar Dubey, and Snigdha Chaturvedi. Towards scalable exact machine unlearning using parameter-efficient fine-tuning. In *International Conference on Learning Representations*, volume 2025, pp. 84753–84779, 2025.

Shristi Das Biswas, Arani Roy, and Kaushik Roy. Cure: Concept unlearning via orthogonal representation editing in diffusion models. *Advances in Neural Information Processing Systems*, 38: 82539–82563, 2026.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, volume 2024, pp. 47922–47945, 2024.

Xinwen Cheng, Zhehao Huang, Wenxing Zhou, Zhengbao He, Ruikai Yang, Yingwen Wu, and Xiaolin Huang. Remaining-data-free machine unlearning by suppressing sample contribution. In *International Conference on Learning Representations*, volume 2026, pp. 27521–27549, 2026.

Ali Ebrahimpour-Boroojeny, Hari Sundaram, and Varun Chandrasekaran. Not all wrong is bad: Using adversarial examples for unlearning. In *International Conference on Machine Learning*, 2025.

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 *International Conference on Learning Representations*, volume 2024, pp. 53643–53673, 2024.

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*, pp. 2426–2436, 2023.

Daiheng Gao, Shilin Lu, Wenbo Zhou, Jiaming Chu, Jie Zhang, Mengxi Jia, Bang Zhang, Zhaoxin Fan, and Weiming Zhang. Eraseanything: Enabling concept erasure in rectified flow transformers. In *International Conference on Machine Learning*, 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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13393–13402. IEEE, 2025.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309. IEEE, 2020.

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. *Advances in Neural Information Processing Systems*, 30, 2017.

Erin Illman and Paul Temple. California consumer privacy act. *The Business Lawyer*, 75(1):1637–1646, 2019.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Kodai Kawamura, Yuta Goto, Rintaro Yanagi, Hirokatsu Kataoka, and Go Irie. Approximate domain unlearning for vision-language models. *Advances in Neural Information Processing Systems*, 38: 21805–21833, 2026.

Shervin Khalafi, Alejandro Ribeiro, and Dongsheng Ding. Unlearning in diffusion models: A unified framework with kl divergence and likelihood constraints. *International Conference on Machine Learning*, 2026.

Miso Kim, Georu Lee, Yunji Kim, Hoki Kim, Jinseong Park, and Woojin Lee. Co-occurring associated retained concepts in diffusion unlearning. *arXiv preprint arXiv:2606.24192*, 2026.

Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. *arXiv preprint arXiv:2506.06985*, 2025.

Nikita Kornilov, Petr Mokrov, Alexander Gasnikov, and Alexander Korotin. Optimal flow matching: Learning straight trajectories in just one step. *Advances in Neural Information Processing Systems*, 37:104180–104204, 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22634–22645. IEEE, 2023.

Naga Sai Abhiram Kusumba, Maitreya Patel, Kyle Min, Changhoon Kim, Chitta Baral, and Yezhou Yang. Eraseflow: Learning concept erasure policies via gflownet-driven alignment. *Advances in Neural Information Processing Systems*, 38:70876–70911, 2026.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Byung Hyun Lee, Sungjin Lim, and Se Young Chun. Localized concept erasure for text-to-image diffusion models using training-free gated low-rank adaptation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18596–18606. IEEE, 2025.

Uichan Lee, Jeonghyeon Kim, and Sangheum Hwang. Localized concept erasure in text-to-image diffusion models via high-level representation misdirection. In *International Conference on Learning Representations*, 2026.

Fengpeng Li, Kemou Li, Qizhou Wang, Bo Han, and Jiantao Zhou. Aegis: Adversarial target-guided retention-data-free robust concept erasure from diffusion models. *International Conference on Machine Learning*, 2026a.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19659–19668. IEEE, 2025.

Jiaqi Li, Qianshan Wei, Chuanyi Zhang, Guilin Qi, Miaozeng Du, Yongrui Chen, Sheng Bi, and Fan Liu. Single image unlearning: Efficient machine unlearning in multimodal large language models. *Advances in Neural Information Processing Systems*, 37:35414–35453, 2024.

Ouxiang Li, Yuan Wang, Xinting Hu, Houcheng Jiang, Yanbin Hao, James Ma, Fuli Feng, et al. Speed: Scalable, precise, and efficient concept erasure for diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 66777–66803, 2026b.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in Neural Information Processing Systems*, 38:40783–40818, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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

Mengyao Lyu, Yuhong Yang, Haiwen Hong, Hui Chen, Xuan Jin, Yuan He, Hui Xue, Jungong Han, and Guiguang Ding. One-dimensional adapter to rule them all: Concepts, diffusion models and erasing applications. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7559–7568. IEEE, 2024.

Fengyuan Miao, Shancheng Fang, Lingyun Yu, Yadong Qu, Yuhao Sun, Xiaorui Wang, and Hongtao Xie. Sderasure: Concept-specific trajectory shifting for concept erasure via adaptive diffusion classifier. In *International Conference on Learning Representations*, 2026.

Saemi Moon, Minjong Lee, Sangdon Park, and Dongwoo Kim. Holistic unlearning benchmark: A multi-faceted evaluation for text-to-image diffusion model unlearning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16356–16366. IEEE, 2025.

Kien Nguyen, Anh Tran, and Cuong Pham. Suma: A subspace mapping approach for robust and effective concept erasure in text-to-image diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19587–19596. IEEE, 2025.

Gaurav Patel and Qiang Qiu. Learning to unlearn while retaining: Combating gradient conflicts in machine unlearning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4211–4221. IEEE, 2025.

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. IEEE, 2022.

Data Protection. General data protection regulation. *Intersoft Consulting, Accessed in October*, 24 (1), 2018.

Mingxing Rao and Daniel Moyer. Generalization and memorization in rectified flow. In *European Conference on Computer Vision*, pp. 537–554. Springer, 2026.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in Neural Information Processing Systems*, 29, 2016.

Qitan Shi, Cheng Jin, Jiawei Zhang, and Yuantao Gu. Retrack: Data unlearning in diffusion models through redirecting the denoising trajectory. In *International Conference on Artificial Intelligence and Statistics*, 2026.

Lorenzo Simone, Davide Bacciu, and Shuangge Ma. Continualflow: Learning and unlearning with neural flow matching. *arXiv preprint arXiv:2506.18747*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Christoforos N Spartalis, Theodoros Semertzidis, Efstratios Gavves, and Petros Daras. Lotus: Large-scale machine unlearning with a taste of uncertainty. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10046–10055. IEEE, 2025.

Kartik Thakral, Tamar Glaser, Tal Hassner, Mayank Vatsa, and Richa Singh. Fine-grained erasure in text-to-image diffusion-based foundation models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9121–9130. IEEE, 2025.

Yujia Tong, Yuze Wang, Jingling Yuan, and Chuang Hu. Robust machine unlearning for quantized neural networks via adaptive gradient reweighting with similar labels. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20603–20612. IEEE, 2025.

Mykola Vysotskyi, Zahar Kohut, Mariia Shpir, Taras Rumezhak, and Volodymyr Karpiv. Critic-guided reinforcement unlearning in text-to-image diffusion. *arXiv preprint arXiv:2601.03213*, 2026.

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

Yongliang Wu, Shiji Zhou, Mingzhuo Yang, Lianzhe Wang, Heng Chang, Wenbo Zhu, Xinting Hu, Xiao Zhou, and Xu Yang. Unlearning concepts in diffusion model via concept domain correction and concept preserving gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8496–8504, 2025b.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yaxin Xiao, Qingqing Ye, Li Hu, Huadi Zheng, Haibo Hu, Zi Liang, Haoyang Li, and Yijie Jiao. Reminiscence attack on residuals: Exploiting approximate machine unlearning for privacy. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3058–3068. IEEE, 2025.

Lexiang Xiong, Liu Chengyu, Jingwen Ye, Yan Liu, and Yuecong Xu. Semantic surgery: Zero-shot concept erasure in diffusion models. *Advances in Neural Information Processing Systems*, 38: 146199–146238, 2026.

Yang Zhang, Er Jin, Yanfei Dong, Yixuan Wu, Philip Torr, Ashkan Khakzar, Johannes Stegmaier, and Kenji Kawaguchi. Minimalist concept erasure in generative models. *International Conference on Machine Learning*, 2025.

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. *Advances in Neural Information Processing Systems*, 37:36748–36776, 2024.

Kairan Zhao, Meghdad Kurmanji, George-Octavian Barbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. *Advances in Neural Information Processing Systems*, 37:12293–12333, 2024.

Zhengyi Zhong, Weidong Bao, Ji Wang, Shuai Zhang, Jingxuan Zhou, Lingjuan Lyu, and Wei Yang Bryan Lim. Unlearning through knowledge overwriting: Reversible federated unlearning via selective sparse adapter. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30661–30670. IEEE, 2025.

APPENDIX

A Notation Table **15**

B Algorithm Details **15**

C Proofs **16**

C.1 Proof of Proposition 3.1 16

C.2 Proof of Proposition 3.2 18

C.3 Proof of Proposition 3.3 19

D Empirical Study **19**

D.1 Dataset Details 19

D.2 Compared Methods 20

D.3 Implementation Details and Experimental Environment 20

E Reward Functions. **21**

F Additional Experimental Results **21**

F.1 Qualitative Analysis on MNIST T-Shirt. 21

F.2 Additional Qualitative Results on CelebA-HQ 22

F.3 Ablation Study 22

F.4 Forgetting–Quality Trade-off 22

A NOTATION TABLE

We provide the important notations and their descriptions for clarification in Table 3.

Table 3: Summary of main notations.

Symbol	Description
$\mathcal{D}$	Original training dataset.
$\mathcal{D}_f, \mathcal{D}_r$	Forget and retain sets.
$p_{\mathcal{D}}$	Empirical distribution over $\mathcal{D}$.
p_0	Initial noise distribution.
x_0, x_1	Noise and data samples.
x_t	Flow state at time t.
v_{θ_0}	Pretrained velocity field.
v_θ	Current student velocity field.
v_ϕ	Frozen teacher velocity field.
θ_0, θ	Pretrained and current parameters.
$\theta_r^\star$	Parameters retrained on $\mathcal{D}_r$.
θ_u	Final unlearned parameters.
p_t	Time-marginal distribution.
σ_t	SDE stochasticity coefficient.
W_t	Standard Wiener process.
t_0	Retain-side rollout start time.
Δt	Transition step size.
$X_{t_0}^r$	Retain-anchored state at t_0.
$\rho_{\theta,t}^r$	Retain-side on-policy state distribution.
$\pi_\theta(\cdot \mid X_t)$	Student transition policy.
$\pi_\phi(\cdot \mid X_t)$	Teacher transition policy.
μ_θ, μ_ϕ	Student and teacher transition means.
$\mathcal{L}_{\mathrm{OPD}}^r$	Retain-side distillation objective.
x_f	Designated forget sample.
τ_i	The i-th sampled trajectory.
G	Number of sampled trajectories.
K	Number of transition steps.
$\hat{x}_i$	Final output of trajectory τ_i.
$\mathcal{F}(\hat{x}_i, x_f)$	Forgetting reward.
$\mathcal{U}_m(\hat{x}_i)$	The m-th quality reward.
M	Number of quality rewards.
a	Reward balancing coefficient.
r_i	Reward of trajectory τ_i.
$\bar{r}, s_r$	Group reward mean and standard deviation.
$\widehat{A}_i$	Group-relative advantage.
θ_{old}	Behavior-policy parameters.
$\rho_{k,i}(\theta)$	Transition importance ratio.
$D_{k,i}^{\mathrm{KL}}$	Forget-side transition KL divergence.
ε	Policy clipping coefficient.
β	KL regularization coefficient.
$\mathcal{J}_f$	Forget-side policy objective.
$\mathcal{L}_{\mathrm{UOPO}}^f$	Forget-side policy loss.
$g_r^{(k)}, g_f^{(k)}$	Retain and forget gradients.
$\lambda_r^{(k)}$	Adaptive retention weight.
ρ	Relative retention strength.
δ	Numerical stability constant.
$\mathcal{L}_{\mathrm{FlowForget}}$	Overall FlowForget objective.

B ALGORITHM DETAILS

Algorithm 1 summarizes the overall training procedure of the **FlowForget**. Algorithm 2 summarizes the unlearning-oriented policy optimization procedure for constructing the forget-side objective.

Algorithm 1 FlowForget

Require: Pretrained model v_{θ_0}, teacher v_ϕ, retain set $\mathcal{D}_r$, forget set $\mathcal{D}_f$, rollout steps K, group size G, retention strength ρ, learning rate η, iterations T
Ensure: Unlearned model v_θ
Initialize $\theta \leftarrow \theta_0$
for $n = 0, \ldots, T-1$ **do**
  Sample retain data from $\mathcal{D}_r$
  Roll out the current model with the marginal-preserving SDE
  Compute transition KLs between π_θ and π_ϕ
  Compute retain objective $\mathcal{L}^r_{\mathrm{OPD}}$
  Sample forget data from $\mathcal{D}_f$
  Compute forget objective $\mathcal{L}^f_{\mathrm{GRPO}}$ using Algorithm 2
  Compute gradients $g_r^{(n)}$ and $g_f^{(n)}$
  Compute adaptive retention weight $\lambda_r^{(n)}$
  Compute joint objective $\mathcal{L}^{(n)}_{\mathrm{FlowForget}}$
  Update model parameters θ
end for
return v_θ

Algorithm 2 Unlearning-Oriented Policy Optimization

Require: Current model v_θ, pretrained model v_{θ_0}, forget set $\mathcal{D}_f$, group size G, rollout steps K, reward weight a, clipping parameter ε, KL coefficient β
Ensure: Forget objective $\mathcal{L}^f_{\mathrm{GRPO}}$
Set behavior-policy parameters $\theta_{\mathrm{old}} \leftarrow \theta$
Sample forget sample $x_f \sim \mathcal{D}_f$
for $i = 1, \ldots, G$ **do**
  Sample stochastic trajectory τ_i from $\pi_{\theta_{\mathrm{old}}}$
  Obtain final output $\hat{x}_i$
  Compute forgetting score $\mathcal{F}(\hat{x}_i, x_f)$
  Compute generation-quality scores $\{\mathcal{U}_m(\hat{x}_i)\}_m$
  Compute unlearning reward $\mathcal{R}_f(\hat{x}_i, x_f)$
end for
Compute group mean and standard deviation of rewards
Compute group-relative advantages $\widehat{A}_i$
for $i = 1, \ldots, G$ **do**
  for $k = 0, \ldots, K-1$ **do**
    Compute transition ratio $\rho_{k,i}(\theta)$
    Compute KL regularization term $D^{\mathrm{KL}}_{k,i}$
    Compute clipped surrogate $S_{k,i}$
  end for
end for
Compute forget-side objective $\mathcal{L}^f_{\mathrm{GRPO}}$
return $\mathcal{L}^f_{\mathrm{GRPO}}$

C PROOFS

C.1 PROOF OF PROPOSITION 3.1

Proposition 3.1 (Marginal-Preserving Stochastic Flow) *For the linear flow-matching path $X_t = (1-t)X_0 + tX_1$, let p_t denote its time-marginal distribution. For a velocity field satisfying the flow-matching optimum $v_\theta(x,t) = \mathbb{E}[X_1 - X_0 \mid X_t = x]$, the flow ODE $dX_t = v_\theta(X_t, t)dt$ can be associated with the following marginal-preserving stochastic counterpart for $t \in (0,1)$:*

$$dX_t = \left[v_\theta(X_t, t) - \frac{\sigma_t^2}{2(1-t)} \left(X_t - t v_\theta(X_t, t) \right) \right] dt + \sigma_t dW_t, \tag{14}$$

where W_t is a standard Wiener process and $\sigma_t = \kappa\sqrt{(1-t)/t}$ controls stochasticity. Under this condition, the SDE and flow ODE share the same time-marginal distribution p_t for $t \in (0,1)$.

Proof. Consider the linear flow-matching path

$$X_t = (1-t)X_0 + tX_1, \tag{15}$$

where $X_0 \sim \mathcal{N}(0, I)$ is independent of X_1, and let p_t denote the marginal density of X_t. Conditioned on X_1, we have

$$p_t(x \mid X_1) = \mathcal{N}\big(tX_1, (1-t)^2 I\big)\,, \tag{16}$$

whose score is

$$\nabla_x \log p_t(x \mid X_1) = -\frac{x - tX_1}{(1-t)^2}. \tag{17}$$

Using the conditional-score identity gives

$$\begin{aligned}\nabla_x \log p_t(x) &= \mathbb{E}[\nabla_x \log p_t(x \mid X_1) \mid X_t = x] \\ &= -\frac{1}{(1-t)^2}\left(x - t\,\mathbb{E}[X_1 \mid X_t = x]\right).\end{aligned} \tag{18}$$

Under the flow-matching optimum,

$$v_\theta(x,t) = \mathbb{E}[X_1 - X_0 \mid X_t = x]\,. \tag{19}$$

Since

$$X_0 = \frac{X_t - tX_1}{1-t}, \tag{20}$$

we obtain

$$v_\theta(x,t) = \frac{\mathbb{E}[X_1 \mid X_t = x] - x}{1-t}, \tag{21}$$

and hence

$$\mathbb{E}[X_1 \mid X_t = x] = x + (1-t)v_\theta(x,t). \tag{22}$$

Substituting Eq. (22) into Eq. (18) yields

$$\begin{aligned}\nabla_x \log p_t(x) &= -\frac{x - t\,[x + (1-t)v_\theta(x,t)]}{(1-t)^2} \\ &= -\frac{x - tv_\theta(x,t)}{1-t}.\end{aligned} \tag{23}$$

The marginal velocity field v_θ induces the continuity equation

$$\partial_t p_t = -\nabla_x \cdot (p_t v_\theta)\,. \tag{24}$$

Now consider an SDE

$$dX_t = b_t(X_t)\,dt + \sigma_t\,dW_t, \tag{25}$$

where σ_t depends only on time. Its marginal density satisfies the Fokker–Planck equation

$$\partial_t p_t = -\nabla_x \cdot (p_t b_t) + \frac{\sigma_t^2}{2}\Delta p_t. \tag{26}$$

Choose

$$b_t(x) = v_\theta(x,t) + \frac{\sigma_t^2}{2}\nabla_x \log p_t(x). \tag{27}$$

Substituting Eq. (27) into Eq. (26) gives

$$\partial_t p_t = -\nabla_x \cdot (p_t v_\theta) - \frac{\sigma_t^2}{2}\nabla_x \cdot (p_t \nabla_x \log p_t) + \frac{\sigma_t^2}{2}\Delta p_t. \tag{28}$$

Since

$$p_t \nabla_x \log p_t = \nabla_x p_t, \tag{29}$$

we have

$$\nabla_x \cdot (p_t \nabla_x \log p_t) = \Delta p_t. \tag{30}$$

Therefore, the last two terms cancel and

$$\partial_t p_t = -\nabla_x \cdot (p_t v_\theta), \tag{31}$$

which is identical to the continuity equation in Eq. (24). Thus, for $t \in (0, 1)$, the stochastic dynamics preserve the same marginal path p_t as the deterministic flow ODE.

Finally, substituting the score–velocity relation Eq. (23) into Eq. (27) yields

$$b_t(x) = v_\theta(x, t) - \frac{\sigma_t^2}{2(1-t)} \left(x - t v_\theta(x, t)\right). \tag{32}$$

Hence,

$$dX_t = \left[v_\theta(X_t, t) - \frac{\sigma_t^2}{2(1-t)} \left(X_t - t v_\theta(X_t, t)\right) \right] dt + \sigma_t dW_t, \tag{33}$$

which is exactly Eq. (3) and completes the proof. □

C.2 Proof of Proposition 3.2

Proposition 3.2 (Gaussian Transition Policy) *Given the Euler–Maruyama discretization of Eq. (3) with $t \in (0, 1)$, $\Delta t > 0$, and $t + \Delta t \leq 1$, the transition $X_t \to X_{t+\Delta t}$ is isotropic Gaussian:*

$$\begin{aligned} &\pi_\theta(X_{t+\Delta t} \mid X_t) = \mathcal{N}\big(\mu_\theta(X_t, t), \sigma_t^2 \Delta t I\big), \\ &\mu_\theta(X_t, t) = X_t + \left[v_\theta(X_t, t) - \frac{\sigma_t^2}{2(1-t)} \big(X_t - t v_\theta(X_t, t)\big) \right] \Delta t. \end{aligned} \tag{34}$$

Proof. Recall the marginal-preserving SDE,

$$dX_t = b_\theta(X_t, t)\, dt + \sigma_t\, dW_t, \tag{35}$$

where

$$b_\theta(X_t, t) = v_\theta(X_t, t) - \frac{\sigma_t^2}{2(1-t)} \left(X_t - t v_\theta(X_t, t)\right). \tag{36}$$

Applying the Euler–Maruyama discretization with step size Δt gives

$$X_{t+\Delta t} = X_t + b_\theta(X_t, t)\Delta t + \sigma_t \Delta W_t, \tag{37}$$

where the Brownian increment satisfies

$$\Delta W_t = W_{t+\Delta t} - W_t \sim \mathcal{N}(0, \Delta t I). \tag{38}$$

Equivalently, we can write

$$\Delta W_t = \sqrt{\Delta t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{39}$$

Conditioned on X_t, the drift term in Eq. (37) is deterministic, while the stochastic term is Gaussian. Therefore, the one-step transition of the discretized process is

$$X_{t+\Delta t} \mid X_t \sim \mathcal{N}\big(X_t + b_\theta(X_t, t)\Delta t,\ \sigma_t^2 \Delta t I\big). \tag{40}$$

Defining

$$\mu_\theta(X_t, t) = X_t + b_\theta(X_t, t)\Delta t \tag{41}$$

and substituting Eq. (36) yields

$$\mu_\theta(X_t, t) = X_t + \left[v_\theta(X_t, t) - \frac{\sigma_t^2}{2(1-t)} \left(X_t - t v_\theta(X_t, t)\right) \right] \Delta t. \tag{42}$$

Hence,

$$\pi_\theta(X_{t+\Delta t} \mid X_t) = \mathcal{N}\big(\mu_\theta(X_t, t), \sigma_t^2 \Delta t I\big), \tag{43}$$

Therefore, we conclude the proof of Proposition 3.2. □

C.3 Proof of Proposition 3.3

Proposition 3.3 (Velocity-Space KL Reduction) *The Gaussian transition policies induced by v_θ and v_ϕ at X_t share covariance $\sigma_t^2 \Delta t I$. Their transition-level KL divergence is given by*

$$D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid X_t) \,\|\, \pi_\phi(\cdot \mid X_t)\right) = \frac{\Delta t}{2}\left(\frac{1}{\sigma_t} + \frac{\sigma_t t}{2(1-t)}\right)^2 \|v_\theta(X_t,t) - v_\phi(X_t,t)\|_2^2 . \tag{44}$$

Proof. From Proposition 3.2, the transition policies induced by v_θ and v_ϕ at the same state X_t are

$$\pi_\theta(\cdot \mid X_t) = \mathcal{N}\big(\mu_\theta(X_t,t), \sigma_t^2 \Delta t I\big), \qquad \pi_\phi(\cdot \mid X_t) = \mathcal{N}\big(\mu_\phi(X_t,t), \sigma_t^2 \Delta t I\big). \tag{45}$$

Thus, the two Gaussian distributions share the covariance

$$\Sigma_t = \sigma_t^2 \Delta t I. \tag{46}$$

For two Gaussian distributions with identical covariance Σ_t, their KL divergence reduces to

$$D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid X_t) \,\|\, \pi_\phi(\cdot \mid X_t)\right) = \frac{1}{2}\left(\mu_\theta - \mu_\phi\right)^\top \Sigma_t^{-1}\left(\mu_\theta - \mu_\phi\right). \tag{47}$$

Since $\Sigma_t^{-1} = (\sigma_t^2 \Delta t)^{-1} I$, Eq. (47) becomes

$$D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid X_t) \,\|\, \pi_\phi(\cdot \mid X_t)\right) = \frac{1}{2\sigma_t^2 \Delta t}\|\mu_\theta(X_t,t) - \mu_\phi(X_t,t)\|_2^2 . \tag{48}$$

From Eq. (4), the transition mean induced by a velocity field v_ψ, with $\psi \in \{\theta, \phi\}$, is

$$\mu_\psi(X_t,t) = X_t + \left[v_\psi(X_t,t) - \frac{\sigma_t^2}{2(1-t)}\left(X_t - t v_\psi(X_t,t)\right)\right]\Delta t. \tag{49}$$

Taking the difference between the two means cancels all terms depending only on X_t, yielding

$$\begin{aligned}\mu_\theta - \mu_\phi &= \left[v_\theta - v_\phi + \frac{\sigma_t^2 t}{2(1-t)}\left(v_\theta - v_\phi\right)\right]\Delta t \\ &= \left(1 + \frac{\sigma_t^2 t}{2(1-t)}\right)\left(v_\theta - v_\phi\right)\Delta t,\end{aligned} \tag{50}$$

where the arguments (X_t, t) are omitted for brevity.

Substituting Eq. (50) into Eq. (48) gives

$$\begin{aligned}D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid X_t) \,\|\, \pi_\phi(\cdot \mid X_t)\right) &= \frac{\Delta t}{2\sigma_t^2}\left(1 + \frac{\sigma_t^2 t}{2(1-t)}\right)^2 \|v_\theta(X_t,t) - v_\phi(X_t,t)\|_2^2 \\ &= \frac{\Delta t}{2}\left(\frac{1}{\sigma_t} + \frac{\sigma_t t}{2(1-t)}\right)^2 \|v_\theta(X_t,t) - v_\phi(X_t,t)\|_2^2 .\end{aligned} \tag{51}$$

Therefore, we conclude the proof of Proposition 3.3. □

D Empirical Study

D.1 Dataset Details

We conduct experiments on image datasets including *MNIST T-Shirt* (LeCun et al., 1998; Xiao et al., 2017), *CelebA-HQ* (Karras et al., 2017), and *CIFAR-10* (Krizhevsky et al., 2009), covering diverse image domains and varying visual complexity, with detailed statistics summarized in Table 4. Specifically, (1) *MNIST T-Shirt* is constructed from MNIST[1], which contains handwritten digits from classes 0–9, and Fashion-MNIST[2], which contains ten categories of fashion products. Following

[1] https://huggingface.co/datasets/ylecun/mnist

[2] https://github.com/zalandoresearch/fashion-mnist

Table 4: Dataset statistics and data-unlearning configurations.

Benchmark	Source Dataset(s)	Native Resolution	Modality	Original Data Size
MNIST T-Shirt	MNIST + Fashion-MNIST	28×28	Grayscale	60K train / 10K test
CelebA-HQ	CelebA-HQ	1024×1024	RGB	30K
CIFAR-10	CIFAR-10	32×32	RGB	50K train / 10K test

SISS (Alberti et al., 2025), we repeatedly inject a single T-shirt image selected from Fashion-MNIST into the MNIST training set, where MNIST forms the retain distribution and the selected T-shirt serves as the designated forget sample, accounting for approximately 2% of the training set. (2) *CelebA-HQ*[3] is a high-quality face dataset derived from CelebA, where designated training images are selected as the forget set and the remaining images form the retain set. (3) *CIFAR-10*[4] is a natural image dataset containing ten object categories, on which we perform sample-level data unlearning by selecting designated training images as the forget set and using the remaining training images as the retain set. The same forget samples are used across all compared methods.

D.2 Compared Methods

We compare our proposed **FlowForget** with a set of representative unlearning baselines: (1) **Pretrained** denotes the original model before unlearning and serves as a reference for evaluating forgetting effectiveness and generation-quality changes. (2) **KL-constrained** (Khalafi et al., 2026) formulates data unlearning as a forward KL-constrained optimization problem that dynamically reweights forget samples according to their unlearning difficulty while limiting deviation from the pretrained model. (3) **ReTrack** (Shi et al., 2026) redirects the generative trajectories of forget samples toward their k-nearest retained neighbors while regularizing the model to preserve the original generative distribution. (4) **SISS** (Alberti et al., 2025) combines retention and subtraction-based forgetting objectives through importance sampling to balance sample removal and generation quality. (5) **EraseDiff** (Wu et al., 2025a) performs constrained unlearning by driving model predictions on forget samples away from their original training targets while preserving utility on retained data. (6) **NegGrad** (Golatkar et al., 2020) performs gradient ascent on the training loss of forget samples, directly pushing the model away from their original training targets.

D.3 Implementation Details and Experimental Environment

Pretrained Models. For *MNIST T-Shirt*, the teacher and student use UNet architectures with channel configurations of $[64, 128]$ and $[32, 64]$, respectively. For *CelebA-HQ* and *CIFAR-10*, both models are initialized from the publicly released pretrained checkpoints of the official Rectified Flow implementation[5] (Liu et al., 2022), with the teacher kept frozen during unlearning.

Additional Training Details. We perform full-parameter optimization of the student while keeping the teacher fixed throughout unlearning. We use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The numerical stability constant in the adaptive retention weight is set to $\delta = 10^{-6}$, and the constant used for group-relative advantage normalization is set to $\epsilon_{\text{num}} = 10^{-6}$. For the stochastic rollout, we evaluate the SDE noise schedule using numerically clipped time values $t \in [10^{-3}, 1 - 10^{-3}]$ to avoid instability near the endpoints. For evaluation, we use a 32-step Euler solver on *MNIST T-Shirt* and a 100-step RK4 solver on both *CelebA-HQ* and *CIFAR-10*. We repeat each experiment over 10 independent runs with different random seeds and report the average results.

Experimental Environment. All experiments are conducted on Linux servers equipped with four NVIDIA A100-SXM4 GPUs with 40 GiB memory each. All code is implemented in Python 3.10.20 under a consistent software environment to ensure reproducibility across experimental runs.

[3] https://huggingface.co/datasets/korexyz/celeba-hq-256x256

[4] https://www.cs.toronto.edu/~kriz/cifar.html

[5] https://github.com/gnobitab/RectifiedFlow

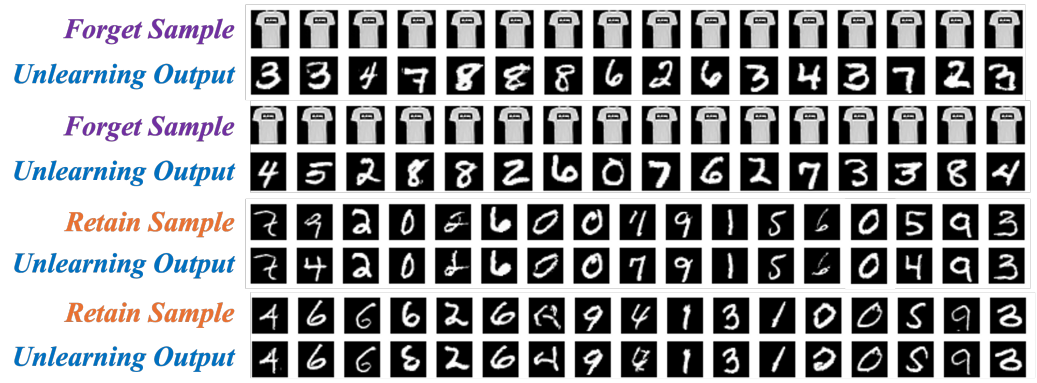


Figure 5: Qualitative comparison before and after unlearning on the MNIST T-Shirt dataset. Each column uses an identical initial noise sample to facilitate a direct visual comparison, with the pre-unlearning output shown above and the post-unlearning output shown below.

E Reward Functions.

Following Eq. (7), we specify the forgetting reward $\mathcal{F}$ and quality-preserving rewards $\mathcal{U}m$.

(1) For *MNIST T-Shirt*, we set $a = 1$ in Eq. (7) and use no additional quality-preserving reward. Since MNIST contains simple and low-dimensional image structures, the pixel-space ℓ_2 distance provides a direct forgetting signal, while generation quality is already preserved by trajectory-aligned distillation on retained samples. We therefore define the forgetting reward as

$$\mathcal{F}(\hat{x}, x_f) = \|\hat{x} - x_f\|_2. \tag{52}$$

A larger $\mathcal{F}$ indicates a greater deviation from the designated T-shirt in pixel space, encouraging the model to reduce its resemblance to the forget sample throughout the generation process.

(2) For *CelebA-HQ*, we instantiate the forgetting reward as

$$\mathcal{F}(\hat{x}_i, x_f) = 1 - \frac{1}{2}\left(\mathcal{S}_{\mathrm{RSSCD}}(\hat{x}_i, x_f) + \mathcal{S}_{\mathrm{Arc}}(\hat{x}_i, x_f)\right), \tag{53}$$

where $\mathcal{S}_{\mathrm{RSSCD}}$ measures robust instance-level similarity and $\mathcal{S}_{\mathrm{Arc}}$ measures identity similarity. Thus, a larger $\mathcal{F}$ indicates that the generated output is less similar to the forget sample at both the instance and identity levels. We further employ DeQA-Score as the quality reward,

$$\mathcal{U}(\hat{x}_i) = \frac{\mathcal{Q}_{\mathrm{DeQA}}(\hat{x}_i) - 1}{4}, \tag{54}$$

where $\mathcal{Q}_{\mathrm{DeQA}} \in [1, 5]$ and a larger value indicates higher perceptual quality. In Eq. (7), we set $a = 0.7$, placing greater emphasis on forgetting while still preserving generation quality.

(3) For *CIFAR-10*, we set $a = 1$ in Eq. (7) and use no additional quality-preserving reward. Since CIFAR-10 consists of low-resolution 32×32 images, the pixel-space ℓ_2 distance provides a direct forgetting signal, while generation quality is already preserved by trajectory-aligned distillation on retained samples. We therefore define the forgetting reward as

$$\mathcal{F}(\hat{x}, x_f) = \|\hat{x} - x_f\|_2. \tag{55}$$

A larger $\mathcal{F}$ indicates a greater deviation from the designated forget sample in pixel space, encouraging the model to reduce its resemblance to the forget sample throughout the generation process.

F Additional Experimental Results

F.1 Qualitative Analysis on MNIST T-Shirt.

As shown in Fig. 5, **FlowForget** effectively changes the outputs associated with the forget sample from the target T-shirt to diverse handwritten digits under the same initial noise. In contrast, outputs

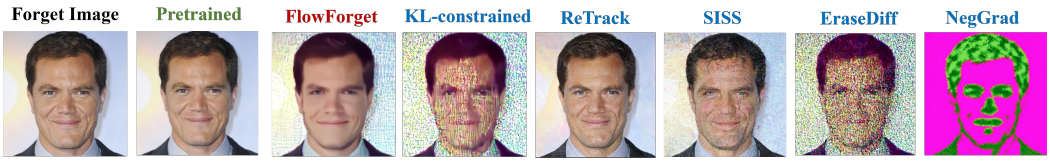


Figure 6: Additional qualitative comparison on *CelebA-HQ*.

Table 5: Ablation results on *MNIST T-Shirt*.

Variant	NLL↑	p (%)↓	IS↑
Pretrained	0.934 ± 0.061	1.6862 ± 0.0743	9.648 ± 0.026
w/o TA-OPD	9.417 ± 0.284	0.0000 ± 0.0000	7.836 ± 0.173
w/o UOPO	1.086 ± 0.093	1.6951 ± 0.0865	9.571 ± 0.064
FlowForget	8.793 ± 0.092	0.0000 ± 0.0000	9.529 ± 0.071

for retained samples remain largely consistent before and after unlearning, with their digit identities and overall structures well preserved. These results demonstrate that **FlowForget** selectively removes the influence of the forget sample while preserving retained generation behavior without introducing noticeable degradation to the remaining data distribution.

F.2 Additional Qualitative Results on CelebA-HQ

Fig. 6 provides additional qualitative comparisons on *CelebA-HQ*. The pretrained model closely reproduces the designated forget image, indicating its persistent influence on the generation process. ReTrack and SISS preserve realistic facial structures but retain noticeable visual similarity to the forget sample, while KL-constrained, EraseDiff, and NegGrad introduce varying degrees of artifacts or distributional distortion. In contrast, **FlowForget** produces a visually distinct yet realistic face, demonstrating effective forgetting while consistently preserving overall generation quality.

F.3 Ablation Study

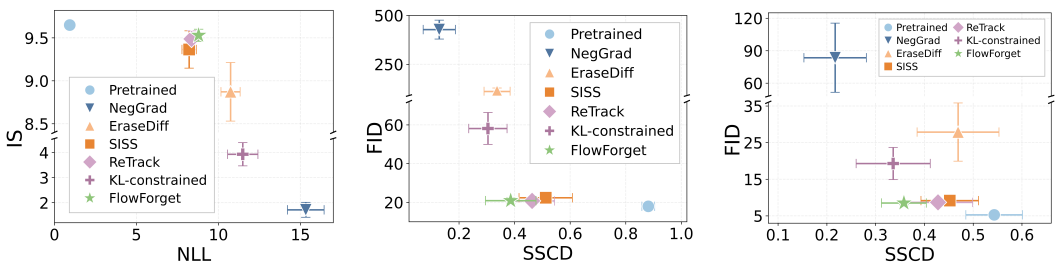
We conduct ablation studies on *MNIST T-Shirt*, *CelebA-HQ*, and *CIFAR-10* to assess the contributions of Trajectory-Aligned On-Policy Distillation (TA-OPD) and Unlearning-Oriented Policy Optimization (UOPO). The results are reported in Tables 5 and 6. Removing TA-OPD retains strong forgetting performance but consistently degrades generation quality, as indicated by the reduced IS on *MNIST T-Shirt* and increased FID on *CelebA-HQ* and *CIFAR-10*. In contrast, removing UOPO preserves generation quality but fails to forget the designated samples, with NLL and SSCD close to the pretrained model. The full **FlowForget** achieves effective unlearning while preserving generation quality, demonstrating the complementary roles of the two components.

Table 6: Ablation results on *CelebA-HQ* and *CIFAR-10*.

Variant	CelebA-HQ			CIFAR-10		
	NLL↑	SSCD↓	FID↓	NLL↑	SSCD↓	FID↓
Pretrained	1.291 ± 0.128	0.881 ± 0.023	17.984 ± 0.436	3.224 ± 0.361	0.543 ± 0.059	5.273 ± 0.214
w/o TA-OPD	2.084 ± 0.319	0.294 ± 0.072	46.731 ± 6.284	3.842 ± 0.376	0.287 ± 0.068	16.943 ± 3.225
w/o UOPO	1.327 ± 0.145	0.874 ± 0.047	19.263 ± 0.817	3.267 ± 0.342	0.537 ± 0.061	6.148 ± 0.436
FlowForget	1.547 ± 0.236	0.386 ± 0.091	20.918 ± 1.642	3.467 ± 0.325	0.358 ± 0.046	8.512 ± 0.488

F.4 Forgetting–Quality Trade-off

Fig. 7 further illustrates the relationship between unlearning effectiveness and generation quality on the three datasets. On *MNIST T-Shirt*, NegGrad, EraseDiff, and KL-constrained achieve stronger forgetting but suffer noticeable quality degradation, whereas SISS and ReTrack preserve high IS



(a) *MNIST T-Shirt* (b) *CelebA-HQ* (c) *CIFAR-10*

Figure 7: Unlearning effectiveness versus generation quality on *MNIST T-Shirt*, *CelebA-HQ*, and *CIFAR-10*. Error bars denote standard deviations across evaluation runs.

with relatively weaker forgetting. **FlowForget** achieves stronger forgetting than SISS and ReTrack while maintaining generation quality close to the pretrained model. Similar observations hold on *CelebA-HQ* and *CIFAR-10*: aggressive forgetting methods obtain stronger unlearning at the cost of substantially increased FID, while preservation-oriented methods retain lower FID but show weaker forgetting. In contrast, **FlowForget** remains in the low-FID region while achieving consistently stronger forgetting, indicating that it reduces the influence of forget samples without substantially disturbing the learned generation distribution. The error bars further show that this behavior remains relatively stable across evaluation runs. Overall, these results confirm that the two proposed components jointly support effective unlearning while preserving overall generation quality.