REDISTRIBUTIVE UNLEARNING FOR ONE-STEP GENERATORS VIA UNBALANCED OPTIMAL TRANSPORT

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in one-step generative frameworks, such as flow map models, have significantly improved the efficiency of image generation by learning direct noise-to-data mappings in a single forward pass. However, machine unlearning for these powerful generators remains entirely unexplored, as existing diffusion unlearning methods rely on multi-step denoising and are inherently incompatible with one-step models. Moreover, prior methods focus only on suppressing the target concept, often leaving the unlearned probability mass to collapse into noise-like samples. In this work, we propose *UOT-Unlearn*, a redistributive class unlearning framework for one-step generative models based on the Unbalanced Optimal Transport (UOT). Our method formulates unlearning as a principled trade-off between a forget cost, which suppresses the target class, and an f-divergence penalty, which confines the generated distribution to the data support via relaxed marginal constraints. As a result, the probability mass of the unlearned class is smoothly redistributed to the remaining classes rather than discarded, preserving overall generation fidelity. Experiments on CIFAR-10 and ImageNet-256 show that UOT-Unlearn achieves superior unlearning performance while maintaining generation quality, and it further extends to one-step text-to-image generation.

1 INTRODUCTION

Generative models, particularly diffusion models, have achieved high-quality image synthesis (Ho et al., 2020; Song et al., 2021b;a). However, their practical utility is heavily bottlenecked by slow inference speeds, which stem from the requirement of tens to hundreds of iterative denoising steps. To overcome this limitation, recent advances have rapidly shifted towards one-step generative architectures, such as consistency models or flow maps (Lipman et al., 2023; Liu et al., 2023; Song et al., 2023; Albergo & Vanden-Eijnden, 2023). By directly mapping the noise distribution to the data distribution in a single forward pass, these models achieve near-diffusion-level generation quality with an advantage in sampling speed.

As these generative models grow faster and more powerful, the risk of producing undesirable content, such as Not Safe For Work (NSFW) imagery or copyrighted materials, has simultaneously amplified. To mitigate these risks without the prohibitive cost of retraining models from scratch, machine unlearning has emerged as an essential safeguard (Bourtoule et al., 2021; Nguyen et al., 2025). While unlearning techniques have been actively developed for standard multi-step diffusion models, the field of one-step generative models remains entirely unexplored. This gap is particularly concerning because the extreme generation speed of one-step models can drastically accelerate the spread of harmful content. Therefore, establishing an unlearning framework for one-step generators needs to be investigated.

Crucially, existing diffusion unlearning methods (Kumari et al., 2023; Zhang et al., 2024; Fan et al., 2024; Gandikota et al., 2023) cannot be straightforwardly applied to one-step frameworks. Previous techniques rely on multi-step denoising processes, i.e., steering noise predictions or gradients at intermediate timesteps. In contrast, one-step models map noise to data in a single forward pass. Due to the dependence on intermediate steps, existing step-by-step modifications are not well-defined in one-step generators (Geng et al., 2025).

To bridge this critical gap, we propose ***UOT-Unlearn***, the first plug-and-play class unlearning framework well-aligned with one-step generative models. We first view the unlearning problem by ex-

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.

ploiting the Unbalanced Optimal Transport (UOT). Unlike the standard Optimal Transport (OT), which enforces strict distribution matching, UOT relaxes the marginal constraints and instead minimizes a trade-off between the transport cost and distributional deviation. This flexibility enables us to control the balance between removing the forget class and preserving the overall distribution.

We introduce an unlearning cost that penalizes samples belonging to the forget concept. This heavy penalization induces distribution mismatch for the forget concept, which leads to the unlearning. Based on the neural optimal transport formulation for UOT, we derive the unlearning objective that fine-tunes one-step generative models. Importantly, UOT-Unlearn operates using only generated samples and a forget centroid, eliminating the need for real retain datasets while preserving the generation quality.

- We introduce UOT-Unlearn, the first unlearning framework tailored for one-step generative models based on the optimal transport formulation.
- We formulate a novel UOT-based objective that smoothly redistributes the target class probability into the remaining classes via an f-divergence penalty.
- Experiments on image datasets (e.g., CIFAR-10, ImageNet-256) using representative one-step models, such as CTM and Meanflow, demonstrate that our method achieves superior class unlearning while mitigating quality degradation. We further validate its applicability to text-to-image generation using SDXL-Turbo.

2 PRELIMINARIES

2.1 ONE-STEP GENERATION VIA PROBABILITY FLOW

Continuous-time generative models, such as diffusion models (Song et al., 2021b) and flow matching (Lipman et al., 2023; Liu et al., 2023), aim to learn a continuous transformation between a tractable noise distribution p_0 and a target data distribution $p_1 = p_{\text{data}}$. This transformation is represented by the Ordinary Differential Equation (ODE) modeling the probability flow:

$$\mathrm{d}x_t = v_\theta(x_t, t)\mathrm{d}t, \quad t \in [0, 1], \tag{1}$$

where v_θ denotes a time-dependent velocity field. In diffusion models, v_θ is implicitly defined via the score function (PF-ODE) (Song et al., 2021b). In flow matching, it is learned through a direct regression to the conditional vector field. Generating samples from these models requires numerically integrating this ODE from $t = 0$ to $t = 1$. This iterative process requires tens to hundreds of neural network evaluations, making inference computationally expensive.

To overcome this limitation, recent advances in Flow Map generative modeling (Geng et al., 2025; Song et al., 2023; Kim et al., 2024) aim to directly learn the solution to the probability flow (Eq. 1), thereby enabling one-step or few-step generation. Formally, models like Consistency Trajectory Model (CTM) (Kim et al., 2024) and MeanFlow (Geng et al., 2025) distill or parameterize the flow map $\psi(x_t, t, s)$, which maps a state at time t directly to time s:

$$\psi(x_t, t, s) = x_t + \int_t^s v_\theta(x_\tau, \tau)\mathrm{d}\tau. \tag{2}$$

By learning this mapping directly, the entire generation process can be executed in a single forward pass:

$$x_1 = \psi(x_0, 0, 1) = G_\theta(x_0). \tag{3}$$

Existing machine unlearning methods for generative models are largely designed for multi-step diffusion processes (Kumari et al., 2023; Zhang et al., 2024; Fan et al., 2024; Gandikota et al., 2023), where modifications are applied at intermediate denoising steps. Such approaches are inherently incompatible with one-step generative architectures. UOT-Unlearn addresses this gap by intervening strictly at the final mapping stage $G_\theta(x_0)$. This trajectory-agnostic approach allows our method to be integrated into any pretrained one-step model.

2.2 UNBALANCED OPTIMAL TRANSPORT (UOT)

Optimal Transport (OT) provides a principled framework for finding a cost-efficient mapping that transforms a source distribution μ into a target distribution ν (Villani et al., 2009; Santambrogio,

2015). In the Kantorovich formulation (Kantorovich, 1948), this is defined by searching for a joint probabilistic coupling π that bridges these two distributions while minimizing the total transport cost (Villani et al., 2009):

$$C(\mu, \nu) := \inf_{\pi \in \Pi(\mu,\nu)} \left[\int_{\mathcal{X}\times\mathcal{Y}} c(x, y) \, \mathrm{d}\pi(x, y) \right], \tag{4}$$

where $\Pi(\mu, \nu)$ denotes the set of all joint probability distributions whose marginals π_0 and π_1 must **exactly match** the source μ and target ν, respectively (Villani et al., 2009). Although OT provides a mathematically principled framework for distribution alignment, the strict marginal constraints can make the resulting transport plan overly rigid. In scenarios where probability mass must be removed or redistributed, such as unlearning, this rigidity can be problematic.

Unbalanced Optimal Transport (UOT) addresses this limitation by relaxing the hard marginal constraints through divergence penalties (Chizat et al., 2018; Liero et al., 2018b). Instead of enforcing exact marginal matching, UOT minimizes a **principled trade-off** between the transport cost and the marginal deviation (Balaji et al., 2020; Choi et al., 2023):

$$C_{ub}(\mu, \nu) := \inf_{\pi \in \mathcal{M}_+(\mathcal{X}\times\mathcal{Y})} \left[\int_{\mathcal{X}\times\mathcal{Y}} c(x, y) \, \mathrm{d}\pi(x, y) + D_{\Psi_1}(\pi_0|\mu) + D_{\Psi_2}(\pi_1|\nu) \right], \tag{5}$$

where $\mathcal{M}_+(\mathcal{X} \times \mathcal{Y})$ denotes the set of positive measures defined on $\mathcal{X} \times \mathcal{Y}$, and π_0, π_1 are the marginal distributions of π. The two f-divergences D_Ψ measure the discrepancy between the marginals (π_i) and the corresponding source and target distributions (μ and ν). Formally, for a convex, lower semi-continuous, and non-negative entropy function Ψ, the f-divergence between the marginal π_0 and the source distribution μ is defined as $D_{\Psi_1}(\pi_0|\mu) = \int_{\mathcal{X}} \Psi_1 \left(\frac{\mathrm{d}\pi_0(x)}{\mathrm{d}\mu(x)} \right) \mathrm{d}\mu(x)$ and similarly for $D_{\Psi_2}(\pi_1|\nu)$. Notably, UOT serves as a generalization of OT; if Ψ_1 and Ψ_2 are chosen as convex indicator functions of $\{1\}$, the formulation precisely recovers the classical OT problem as any marginal mismatch results in infinite cost (Choi et al., 2023).

Notably, UOT serves as a generalization of OT; if Ψ_1 and Ψ_2 are chosen as convex indicator functions of $\{1\}$, the formulation precisely recovers the classical OT problem, as any marginal mismatch incurs infinite cost (Choi et al., 2023). This flexibility makes UOT particularly suitable for unlearning, as a suitably designed transport cost can induce marginal mismatch in the forget region.

Choi et al. (2023) proposed a neural optimal transport algorithm, called UOTM, for learning the optimal transport map of the UOT problem. By leveraging the semi-dual form of UOT, UOTM introduces the following learning objective where the transport map T_θ and the dual potential v_ϕ are parameterized by neural networks:

$$\inf_{v_\phi} \left[\int_{\mathcal{X}} \Psi_1^* \left(-\inf_{T_\theta} [c(x, T_\theta(x)) - v_\phi(T_\theta(x))] \right) \mathrm{d}\mu(x) + \int_{\mathcal{Y}} \Psi_2^*(-v_\phi(y)) \, \mathrm{d}\nu(y) \right], \tag{6}$$

where Ψ^* denotes the convex conjugate of Ψ. For instance, when D_{Ψ_1} is the KL divergence, $\Psi_1^*(s) = e^s - 1$. After training, the transport network T_θ learns an unbalanced optimal transport map between the source distribution μ and the target distribution ν.

3 Proposed Method

3.1 Problem Formulation

We consider the problem of **class unlearning for a one-step generative model**. Let $G_{\mathrm{pre}} : \mathcal{Z} \to \mathcal{X}$ denote a pretrained one-step generative model that maps a prior noise distribution $x_0 \sim \mu_Z$ to the learned data distribution $p_{\mathrm{pre}} = (G_{\mathrm{pre}})_{\#}\mu_Z$. The model is trained on the full data distribution p_{data}, which includes both the target concept to be removed (the *forget class*) and all other concepts (the *remaining classes*). Our goal is to fine-tune G_{pre} into a new generator G_θ that removes the forget class while preserving the generation quality and diversity of the remaining classes.

Formally, let $\mathcal{S}_f \subset \mathcal{X}$ denote the semantic support in the data space corresponding to the forget class, and let $\mathcal{S}_r \subset \mathcal{X}$ denote the semantic support for the remaining classes. While conventional

unlearning focuses on suppressing the forget class, our framework achieves a more ambitious goal: ensuring that generated samples avoid $\mathcal{S}_f$ and instead fall within $\mathcal{S}_r$. This objective is formulated as:

$$\mathbb{P}\big(G_\theta(x_0) \in \mathcal{S}_f\big) \to 0, \quad \text{and ideally,} \quad \mathbb{P}\big(G_\theta(x_0) \in \mathcal{S}_r\big) \to 1. \tag{7}$$

Crucially, we achieve this objective for one-step generators operating in a single forward pass—without any access to real retain data during optimization.

3.2 Unlearning via Unbalanced Optimal Transport

UOT Formulation. We formulate the machine unlearning process as a distribution transportation problem using the Unbalanced Optimal Transport (UOT) framework. The key idea is to exploit the intrinsic trade-off between the transportation cost and the distribution matching error. Specifically, by designing a cost function $c_{\text{ul}}(\cdot,\cdot)$ that imposes a heavy penalty on the *forget* region (detailed in Section 3.3), we force the transport plan to avoid generating penalized concepts. To mitigate this massive cost, the UOT objective naturally allows for a distribution matching error, safely steering the generative pathway toward an *unlearned* distribution.

To formalize this framework, we consider the following UOT problem where the source is the pretrained distribution, i.e., $\mu = p_{\text{pre}}$, and the target distribution is the full data distribution $\nu = p_{\text{data}}$.

$$\inf_{\pi \in \mathcal{M}_+(\mathcal{X}\times\mathcal{Y})} \left[\int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\,\mathrm{d}\pi(x,y) + D_{\Psi_1}(\pi_0|p_{\text{pre}}) + D_{\Psi_2}(\pi_1|p_{\text{data}}) \right], \tag{8}$$

Here, the source p_{pre} is the pretrained distribution before unlearning, and the transported marginal π_1 corresponds to the distribution produced by the fine-tuned generator G_θ after unlearning.

The UOT framework induces two desirable properties for π_1. First, the divergence term $D_{\Psi_2}(\pi_1|p_{\text{data}})$ encourages π_1 to remain close to the data distribution, thereby preserving the overall generation fidelity. Second, when the transport cost assigns a large penalty to the forget region $\mathcal{S}_f$, the optimal transport plan avoids placing probability mass in this region. As a result, the generation of the forget class is effectively suppressed, leading to

$$\mathbb{P}\big(G_\theta(x_0) \in \mathcal{S}_f\big) \to 0. \tag{9}$$

At the same time, the divergence regularization ensures that the redistributed probability mass remains within the support of the data distribution. In particular, when D_{Ψ_2} is set to the Kullback–Leibler divergence (as used in our experiments), $D_{\Psi_2}(\pi_1|p_{\text{data}})$ corresponds to a reverse KL divergence, which exhibits the mode-seeking property (Murphy, 2022). This divergence heavily penalizes probability mass assigned to regions where the data distribution has negligible density (Murphy, 2022). In other words, the optimal unlearned distribution $\pi_1^\star$ does not assign positive mass outside $\mathcal{S}_f \cup \mathcal{S}_r$. As a result, the redistributed mass concentrates on valid semantic regions corresponding to the remaining classes, leading to

$$\mathbb{P}\big(G_\theta(x_0) \in \mathcal{S}_r\big) \to 1. \tag{10}$$

Unlearning Objective. Based on this formulation, we derive the unlearning objective using the semi-dual UOT framework. Let ΔT denote the UOT map that transforms the pretrained distribution p_{pre} to the unlearned distribution π_1. Following the UOTM formulation (Eq. 6), we parameterize this transport map by a neural network ΔT_θ and obtain the following learning objective:

$$\inf_{v_\phi} \left[\int_{\mathcal{X}} \Psi_1^*\Big(-\inf_{\Delta T_\theta} \big\{ c(x_1, \Delta T_\theta(x_1)) - v_\phi(\Delta T_\theta(x_1)) \big\} \Big) \mathrm{d}p_{\text{pre}}(x_1) + \int_{\mathcal{Y}} \Psi_2^*(-v_\phi(y))\,\mathrm{d}\nu(y) \right]. \tag{11}$$

To optimize this objective efficiently, we leverage the pushforward structure of the pretrained one-step generator ($p_{\text{pre}} = G_{\text{pre}\#}\mu_Z$). This allows us to reparameterize samples from the generated distribution through the latent variable $x_0 \sim \mu_Z$, where $x_1 = G_{\text{pre}}(x_0)$. Applying this change of variables yields

$$\inf_{v_\phi} \left[\int_{\mathcal{Z}} \Psi_1^*\Big(-\inf_{\Delta T_\theta} \big\{ c(G_{\text{pre}}(x_0), (\Delta T_\theta \circ G_{\text{pre}})(x_0)) - v_\phi((\Delta T_\theta \circ G_{\text{pre}})(x_0)) \big\} \Big) \mathrm{d}\mu_Z(x_0) \right.$$
$$\left. + \int_{\mathcal{Y}} \Psi_2^*(-v_\phi(y))\,\mathrm{d}\nu(y) \right]. \tag{12}$$

By identifying the composed mapping $(\Delta T_\theta \circ G_{\text{pre}})$ with the fine-tuned generator G_θ, we obtain the following unlearning objective:

$$\inf_{v_\phi} \left[\int_{\mathcal{Z}} \Psi_1^*\Big(-\inf_{G_\theta} \big\{ c_{\text{ul}}(G_{\text{pre}}(x_0), G_\theta(x_0)) - v_\phi(G_\theta(x_0)) \big\} \Big) \mathrm{d}\mu_Z(x_0) + \int_{\mathcal{Y}} \Psi_2^*\big(-v_\phi(y) \big) \mathrm{d}\nu(y) \right].$$

Finally, in our constrained unlearning setting where real retain data cannot be accessed, directly evaluating the expectation with respect to the data distribution $\nu = p_{\text{data}}$ is infeasible. To address this issue, we approximate the target distribution using the pretrained distribution itself, i.e., $\nu \approx p_{\text{pre}}$. With pre-computed anchors, subsequent optimization requires no real data while preserving the structural guidance of the original generator (See Algorithm 1 for details).

$$\inf_{v_\phi} \left[\int_{\mathcal{Z}} \Psi_1^*\Big(-\inf_{G_\theta} \big\{ c_{\text{ul}}(G_{\text{pre}}(x_0), G_\theta(x_0)) - v_\phi(G_\theta(x_0)) \big\} \Big) \mathrm{d}\mu_Z(x_0) + \int_{\mathcal{Z}} \Psi_2^*\big(-v_\phi(G_{\text{pre}}(x_0)) \big) \mathrm{d}\mu_Z(x_0) \right]. \quad (13)$$

3.3 Cost Design for Unlearning

To implement the objective in Eq. 13, we introduce the unlearning cost function $c_{\text{ul}}(\cdot, \cdot)$ that explicitly penalizes the generation of the forget class while preserving the remaining concepts. We first compute an *anchor vector* μ_f that represents the center of the forget class in a feature space:

$$\mu_f = \frac{1}{|\mathcal{D}_f|} \sum_{x \in \mathcal{D}_f} f(x), \quad (14)$$

where $\mathcal{D}_f$ is a set of real forget samples and $f(\cdot)$ is a pretrained feature extractor. This anchor provides a compact representation of the forget concept in the feature space.

To identify generated samples in the forget region $\mathcal{S}_f$ (Eq. 7), we define a computable proxy called the *active forget region* $\mathcal{R}_f \subset \mathcal{X}$. Specifically, $\mathcal{R}_f$ consists of generated samples whose feature representations lie within a margin m of the forget anchor μ_f:

$$\mathcal{R}_f = \Big\{ x \in \mathcal{X} \;\Big|\; d_{\cos}\big(f(x), \mu_f\big) < m \Big\}. \quad (15)$$

where $d_{\cos}(a, b) = 1 - \frac{a^\top b}{\|a\|\|b\|}$ denotes the cosine distance, and m is a hyperparameter that defines the semantic boundary of the forget concept in the feature space. During training, generated samples are dynamically checked against this region to identify forget-like outputs.

Unlearning Cost. Based on this region, the unlearning cost is defined as

$$c_{\text{ul}}(G_{\text{pre}}(x_0), G_\theta(x_0)) = \begin{cases} \lambda \cdot \big(m - d_{\cos}(f(G_\theta(x_0)), \mu_f)\big), & \text{if } G_\theta(x_0) \in \mathcal{R}_f, \\ \tau \cdot \|G_{\text{pre}}(x_0) - G_\theta(x_0)\|_2^2, & \text{otherwise.} \end{cases} \quad (16)$$

The two branches of this cost play complementary roles. (i) *Forget cost*: for generated samples inside $\mathcal{R}_f$, the hinge-like penalty pushes their features away from the anchor μ_f, but only until the cosine distance exceeds the margin m. (ii) *Retain cost*: for samples outside $\mathcal{R}_f$, the squared L_2 distance to the pretrained output acts as a transport cost that anchors the remaining classes to their original outputs, while guiding expelled forget samples toward the nearest remaining classes. The weights λ and τ balance active forgetting against fidelity retention. Through this design, our UOT framework safely redistributes the forget mass into retain classes.

3.4 Theoretical Guarantee

We now formalize the unlearning mechanism of UOT-Unlearn. The following proposition shows that the forget weight λ quantitatively controls unlearning, and that the KL penalty constrains the target marginal to the data support.

Table 1: **Comparison of data requirements.** Unlike baselines requiring real data during unlearning, our framework uses forget data only **once** and synthetic samples thereafter. *SalUn utilizes generated data only for specific large-scale tasks. †Our method uses real forget data only once before unlearning fine-tuning to compute the forget centroid μ_f (Eq. 14).

Method	Data Requirements		
	Real Forget	Real Retain	Generated Data
GA	Required	None	None
VDU	Required	None	None
SA	None	None	Required
SalUn	Required	Required	Required*
Ours	**Once†**	**None**	**Required**

Proposition 1 (Informal)**.** *Suppose the forget and retain supports are separated in feature space by a margin $\varepsilon > 0$ around the threshold m, the forget class occupies a fraction $p_f = p_{\mathrm{data}}(\mathcal{S}_f) \in (0,1)$, and $D_{\Psi_2} = D_{\mathrm{KL}}$. Let π_1^λ denote the target marginal of any optimal plan of the UOT problem 8 with the unlearning cost 16. Then*

$$\pi_1^\lambda \ll p_{\mathrm{data}}, \qquad \pi_1^\lambda(\mathcal{S}_f) \;\le\; \frac{C_0}{\varepsilon} \cdot \frac{1}{\lambda}, \tag{17}$$

and $D_{\Psi_2}\big(\pi_1^\lambda \,\big|\, p_{\mathrm{data}}\big) \le C_0$ uniformly in λ, where

$$C_0 \;=\; \tau\, W_2^2\big(p_{\mathrm{pre}},\, p_{\mathrm{data}}(\cdot \mid \mathcal{S}_r)\big) \;+\; \log \tfrac{1}{1-p_f} \tag{18}$$

is a constant independent of λ.

The two conclusions correspond to the two parts of our unlearning goal. The bound in 17 shows that the mass assigned to the forget region vanishes at rate $O(1/\lambda)$, which establishes Eq. 9. The second part follows from absolute continuity: since the KL divergence assigns an infinite penalty to any mass outside $\mathrm{supp}(p_{\mathrm{data}})$, no transport plan with finite objective value can place mass outside $\mathrm{supp}(p_{\mathrm{data}})$. Together with the first bound, this implies that the expelled probability mass is absorbed by the retain support $\mathcal{S}_r$ rather than by off-support, noise-like regions, as stated in Eq. 10. For the formal statement and its proof, see Appendix B.

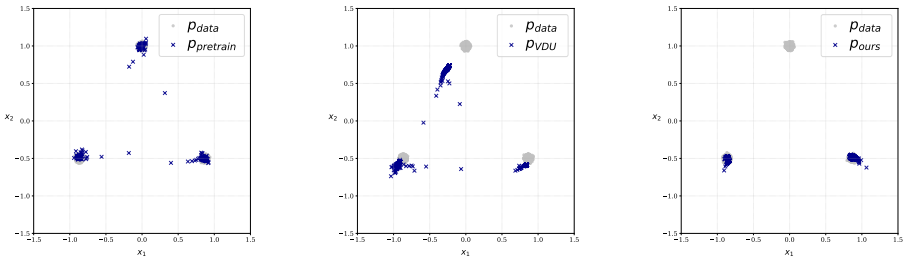
4 Related Works

Existing unlearning methods for generative models are largely designed for diffusion models. As a baseline, Gradient Ascent (GA) (Thudi et al., 2022) applies gradient ascent on the forgetting data. Selective Amnesia (SA) (Heng & Soh, 2023) leverages Elastic Weight Consolidation (EWC) to preserve retained knowledge during unlearning. Saliency Unlearning (SalUn) (Fan et al., 2024) identifies salient parameters to localize forgetting updates. Variational Diffusion Unlearning (VDU) (Panda et al., 2026) formulates unlearning as a variational inference problem to balance forgetting and retention. See Appendix D for details.

Unlike these diffusion-centric approaches, our UOT-based framework serves as a universal unlearning solution tailored for one-step generators. As shown in Table 1, our model-agnostic method overcomes the heavy data reliance of prior works by requiring real forget data only once. Relying purely on synthetic samples thereafter, it enables highly efficient unlearning for one-step generators.

5 Experiments

This section presents a systematic evaluation of our UOT-based unlearning framework to comprehensively validate its core properties. In Section 5.1, we utilize a 2D synthetic dataset to visually analyze the probability redistribution during the unlearning process. In Section 5.2, we evaluate the framework on high-dimensional image benchmarks, including CIFAR-10 (Krizhevsky et al., 2009) and ImageNet-256 (Russakovsky et al., 2015), to assess its scalability and the fundamental trade-off between concept erasure and generative fidelity. In Section 5.3, we evaluate our framework on a text-to-image generation task using SDXL-Turbo (Sauer et al., 2024) as the pretrained backbone. Implementation details and ablation studies are provided in Appendix C, E, and F.



(a) Pretrained (b) VDU (Baseline) (c) **UOT-Unlearn**

Figure 1: **Unlearning results on a 2D toy dataset** with the forget mode at $(0, 1)$. (a) Pretrained one-step generator. (b) VDU leads to overall distribution distortion. (c) Our method redistributes the forget mode to the remaining modes.

Table 2: **Unlearning performance on CIFAR-10 and ImageNet-256.** Best results are highlighted in **bold**, and second-best results are underlined. UOT-Unlearn consistently achieves competitive Percentage of Unlearning (PUL) while preserving the original u-FID (FID of the pretrained generator on the retain classes).

Dataset	Unlearned Class	Original u-FID ↓	GA PUL (%) ↑	GA u-FID ↓	SA PUL (%) ↑	SA u-FID ↓	SalUn PUL (%) ↑	SalUn u-FID ↓	VDU PUL (%) ↑	VDU u-FID ↓	UOT-Unlearn (Ours) PUL (%) ↑	UOT-Unlearn (Ours) u-FID ↓
CIFAR-10	Target Model: Consistency Trajectory Models (CTM) (Kim et al., 2024)											
	Class 1 (Auto)	4.53	88.07	160.03	**91.16**	49.80	90.64	41.36	53.86	71.98	80.32	**9.90**
	Class 6 (Frog)	5.02	39.16	72.17	61.85	42.54	43.16	39.00	51.25	47.11	**90.98**	**5.11**
	Class 8 (Ship)	4.36	**95.40**	208.80	64.50	62.48	48.65	45.70	48.43	52.07	85.23	**5.88**
	Average	4.64	74.21	147.00	72.50	51.61	60.82	42.02	51.18	57.05	**85.51**	**6.96**
	Target Model: MeanFlow (Geng et al., 2025)											
	Class 1 (Auto)	7.73	93.43	115.59	98.05	82.92	**99.35**	62.77	33.70	73.73	90.63	**17.69**
	Class 6 (Frog)	9.23	60.59	47.40	58.49	21.14	50.14	20.06	50.42	66.86	**96.25**	**19.58**
	Class 8 (Ship)	7.08	85.74	49.08	84.98	20.48	78.01	**15.32**	71.47	33.06	**90.43**	21.31
	Average	8.01	79.92	70.69	80.51	41.51	75.83	32.72	51.86	57.88	**92.44**	**19.53**
ImageNet-256	Target Model: MeanFlow (Geng et al., 2025)											
	Class 84	5.53	92.55	55.26	45.80	11.72	50.89	44.09	82.21	127.90	**99.25**	**6.61**
	Class 207	5.50	98.37	111.70	66.56	12.14	60.89	90.04	69.44	67.39	**99.67**	**11.74**
	Class 402	5.52	90.56	85.13	87.83	14.79	61.97	27.49	64.93	87.95	**97.05**	**6.38**
	Class 555	5.53	91.15	46.37	57.72	13.87	50.29	45.22	68.23	79.89	**98.16**	**6.62**
	Class 963	5.50	90.17	41.05	48.54	**10.99**	64.06	42.50	55.41	54.78	**99.63**	11.36
	Average	5.52	92.56	67.90	61.29	12.70	57.62	49.87	68.04	83.58	**98.75**	**8.54**

We compare our approach against the established unlearning baselines: *Gradient Ascent (GA)*, *Selective Amnesia (SA)* (Heng & Soh, 2023), *Saliency Unlearning (SalUn)* (Fan et al., 2024), and *Variational Diffusion Unlearning (VDU)* (Panda et al., 2026). We modified the objectives to operate within a single forward-pass generation framework if needed.

5.1 2D Synthetic Data

We utilize a 2D synthetic dataset consisting of three Gaussian modes to examine the unlearning dynamics in a visually interpretable setting. The primary goal of this experiment is to evaluate whether the probability associated with the forget target is re-assigned to the remaining classes.

Our UOT-based framework enables a principled redistribution of probability mass by leveraging the f-divergence penalty to relax strict marginal constraints. As shown in Figure 1, the probability density previously assigned to the forget mode is smoothly remapped toward the supports of the retain modes. This mechanism ensures that the generator remains within the support of the remaining classes, effectively displacing the targeted concept while preserving the shape and density of the retain distribution, as explained in Section 3.2. In contrast, while the baseline VDU is successful in unlearning the forget modes, the removed probability mass is redistributed into the regions outside the support of p_{data}.

5.2 Image Unlearning Benchmarks

Evaluation Setup and Metrics. First, we conduct experiments on the CIFAR-10 dataset using Consistency Trajectory Models (CTM) (Kim et al., 2024) and MeanFlow (MF) (Geng et al., 2025) as

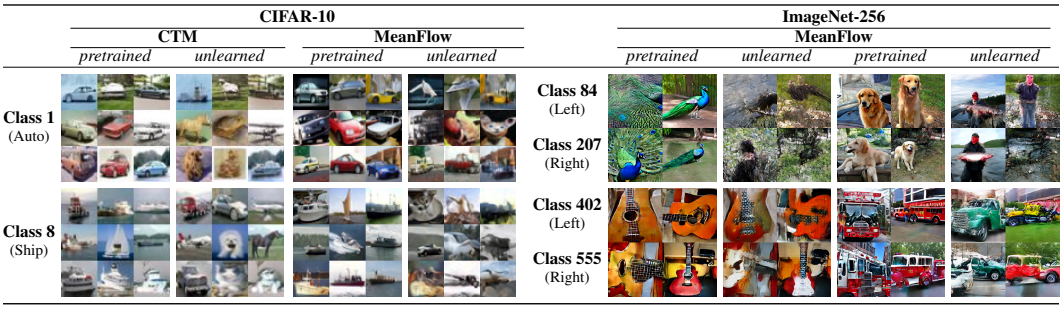


Figure 2: **Qualitative unlearning results on CIFAR-10 and ImageNet-256**. Unconditional samples of CIFAR-10 target classes generated by CTM and MF, and class-conditional samples of ImageNet-256 target classes generated by MF. To illustrate the semantic erasure of targeted concepts, *Unlearned* outputs use the exact same initial noise seeds as their *Pretrained* counterparts.

representative one-step generative architectures. To assess concept erasure, we perform single-class unlearning targeting the same classes as previous works (Panda et al., 2026): classes 1 (*automobile*), 6 (*frog*), and 8 (*ship*).

We further scale UOT-Unlearn to ImageNet-256 using a class-conditional MeanFlow model. Eq. 13 is the original unconditional formulation used for CIFAR-10. For ImageNet-256, we evaluate five forget targets, obtained by randomly selecting five distinct subclasses and then randomly selecting one class from each subclass. We extend the cost with one additional case: forget-class samples outside the forget region $\mathcal{R}_f$ are matched to pretrained samples conditioned on a designated retain class. The full conditional formulation is provided in Appendix D.5.

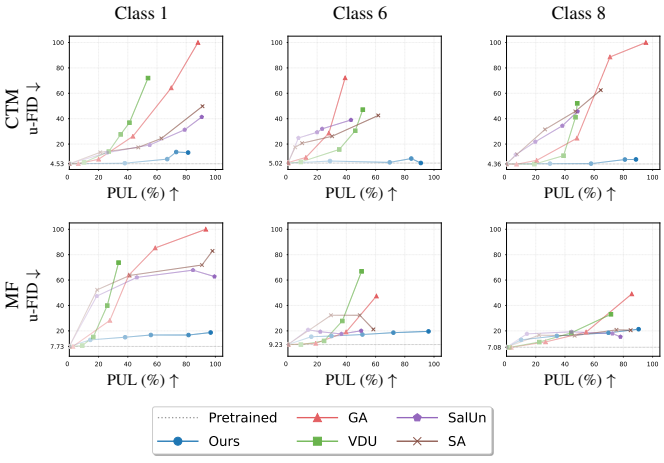


Figure 3: **Step-wise unlearning trajectories on CIFAR-10.** Trade-off between concept erasure (PUL ↑) and generative fidelity (u-FID ↓) for CTM (top) and MF (bottom).

For the feature extractor $f(\cdot)$, we use separate pretrained classifiers to compute the unlearning cost c_{ul}. For the forget anchor μ_f, we use a subset of the forget-class samples (10% for CIFAR-10 and 20% for ImageNet), demonstrating that our method requires minimal data.

To quantitatively evaluate the performance, we employ two primary metrics. First, **Percentage of Unlearning (PUL)** (Tiwary et al., 2025) measures the effectiveness of removing the forget class by computing the relative reduction in its generation frequency. To ensure fair evaluation, we use independent classifiers that are not involved in the training procedure, i.e., in computing the unlearning cost. Second, **Unlearned FID (u-FID)** (Panda et al., 2026) evaluates the generative quality of the retained classes. We compute the Fréchet Inception Distance (FID) (Heusel et al., 2017) between the generated image distributions and the real images restricted to the retain classes. Specifically, on CIFAR-10, we calculate the u-FID between the 45,000 real retain images (the full training set excluding the forget class) and 45,000 images generated by the unlearned model.

Evaluation Results. As summarized in Table 2, our method achieves consistently high PUL scores across all targeted classes while maintaining an exceptionally low u-FID. For instance, on CIFAR-10 with CTM, GA attains an average PUL of 74.21% at the cost of a u-FID of 147.00, whereas our method reaches 85.51% PUL with a u-FID of 6.96, close to the pretrained reference of 4.64. Furthermore, qualitative results in Figure 2 validate that our approach does not merely delete targeted concepts, but effectively maps them into the distributions of remaining classes (qualitative comparisons with the baselines are provided in Appendix G). Finally, Figure 3 highlights the step-wise

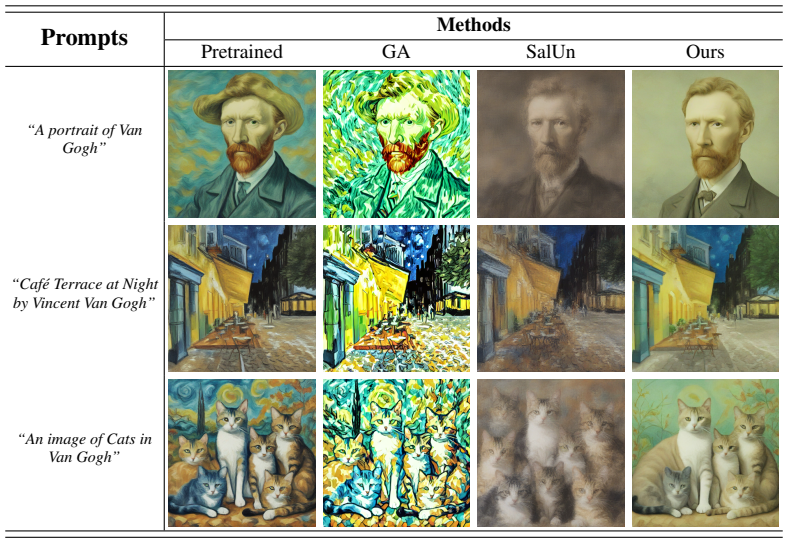


Figure 4: **Qualitative comparison of Van Gogh style erasure on SDXL-Turbo.** Our method effectively removes the target style while generating sharp and natural images.

unlearning dynamics. While higher PUL generally comes with degraded u-FID for the baselines, our method demonstrates a remarkably favorable trajectory, effectively mitigating this trade-off.

5.3 Artistic Style Unlearning

Evaluation Setup and Metrics. We further evaluate UOT-Unlearn on text-to-image (T2I) style unlearning, forgetting the "Van Gogh" style. We adopt a one-step distilled T2I model, SDXL-Turbo (Sauer et al., 2024), as the pretrained backbone. We obtain 90 forget prompts (e.g., "A portrait of Van Gogh") generated by Claude 4.8 (Anthropic, 2026), which include 50 prompts from UD (Zhang et al., 2023). We then construct 90 retain prompts by removing words related to "Van Gogh" from the forget prompts (e.g., "A portrait").

We evaluate style unlearning across two aspects: target-style forgetting and preserving general generation ability. For forgetting, we compute $\text{CLIP}_{\text{style}}$ (CLIP ViT-L/14 (Radford et al., 2021)) between the forget prompts and the corresponding generated images; lower scores indicate better Van Gogh erasure. For preservation, we report $\text{CLIP}_{\text{prompt}}$ (using COCO-30K captions (Lin et al., 2014), text-image alignment).

Table 3: **Unlearning performance on "Van Gogh" style.** $\text{CLIP}_{\text{style}}$ measures forgetting, while $\text{CLIP}_{\text{prompt}}$ (evaluated on COCO-30K) assesses text-image alignment.

Concept	Method	$\text{CLIP}_{\text{style}}\downarrow$	$\text{CLIP}_{\text{prompt}}\uparrow$
Van Gogh	Pretrained	27.39	26.64
	GA	28.26	26.59
	SalUn	23.73	26.60
	Ours	**23.32**	**26.64**

Evaluation Results. Table 3 shows that our method provides a favorable balance between style erasure and generation utility. It achieves the lowest $\text{CLIP}_{\text{style}}$ score among the compared methods, indicating effective removal of the Van Gogh style, while maintaining $\text{CLIP}_{\text{prompt}}$ value comparable to the pretrained model and baselines. The qualitative results in Figure 4 further support this trend: our method suppresses the target style while preserving visually natural and sharp generated images.

6 Conclusion

We introduced UOT-Unlearn, a machine unlearning framework for one-step generative models that casts concept erasure as an unbalanced optimal transport problem. By incorporating an unlearning cost into the single-pass mapping, our method redistributes probability mass away from the forget region without requiring real retain data. Experiments across image and text-to-image generation show that UOT-Unlearn achieves effective concept removal while preserving generation quality. Future work includes exploring structured probability redistribution across related semantic classes within highly structured latent spaces, such as large-scale hierarchical datasets, as well as analyzing the theoretical stability of the unlearning process under alternative cost formulations.

AI USE STATEMENT

In this work, we used generative AI tools for **providing feedback on the experimental design**. We have not used generative AI tools for **formulating mathematical claims**, and **interpreting results** are not applicable to this work. Additionally, we used generative AI tools for **creating and editing software code**. We have reviewed all AI-assisted work. **All AI-assisted outputs were verified and revised by the authors**. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We confirm that we have read and followed the ICLR Code of Ethics. To the best of our knowledge, this work is novel and all relevant prior work has been appropriately cited. All authors have contributed to the work and take responsibility for its content.

REPRODUCIBILITY STATEMENT

We have taken substantial efforts to ensure the reproducibility of our results. Detailed experimental settings for our method and baselines, along with implementation frameworks, are provided in Appendix C. We will also release all code, models to support the reproducibility of our findings.

REFERENCES

Michael Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR 2023 Conference*, 2023.

Anthropic. Claude opus 4.8 system card. Technical report, Anthropic, 2026. URL `https://www.anthropic.com/news/claude-opus-4-8`.

Yogesh Balaji, Rama Chellappa, and Soheil Feizi. Robust optimal transport with applications in generative modeling and domain adaptation. *Advances in Neural Information Processing Systems*, 33:12934–12944, 2020.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021. doi: 10.1109/SP40001.2021.00019.

Yaofo Chen. Pytorch cifar models. `https://github.com/chenyaofo/pytorch-cifar-models`, 2025. Accessed: 2025-5-17.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11): 3090–3123, 2018.

Jaemoo Choi, Jaewoong Choi, and Myungjoo Kang. Generative modeling through the semi-dual formulation of unbalanced optimal transport. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 42433–42455, 2023.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, 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.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025.

Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 17170–17194, 2023.

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 *Advances in neural information processing systems*, volume 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Leonid Vitalevich Kantorovich. On a problem of monge. *Uspekhi Mat. Nauk*, pp. 225–226, 1948.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations*, 2024.

F. Kong et al. An efficient membership inference attack for the diffusion model by proximal initialization. In *ICLR*, 2024.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22634–22645, 10 2023. doi: 10.1109/ICCV51070.2023.02074.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211 (3):969–1117, 2018a.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211 (3):969–1117, 2018b.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Kevin P Murphy. *Probabilistic machine learning: an introduction*. MIT press, 2022.

Thanh Tam Nguyen, Thanh Trung Huynh, Zhao Ren, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5), 2025. doi: 10.1145/3749987.

Subhodip Panda, MS Varun, Shreyans Jain, Sarthak Kumar Maharana, and AP Prathosh. Variational diffusion unlearning: A variational inference framework for unlearning in diffusion models. In *NeurIPS Safe Generative AI Workshop*, 2024.

Subhodip Panda, MS Varun, Shreyans Jain, Sarthak Kumar Maharana, and AP Prathosh. Unlearning in diffusion models under data constraints: A variational inference approach. *Transactions on Machine Learning Research*, 2026.

Huy Phan. Pytorch_cifar10: Pretrained models on cifar-10. https://github.com/huyvnphan/PyTorch_CIFAR10/tree/v3.0.1, 2021. v3.0.1.

PyTorch Contributors. Torchvision: Models and pre-trained weights. https://docs.pytorch.org/vision/0.14/models.html, 2023. Accessed: 2026-06-10.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Filippo Santambrogio. *Optimal transport for applied mathematicians*. Springer, 2015.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 303–319. IEEE, 2022.

Piyush Tiwary, Atri Guha, Subhodip Panda, and Prathosh AP. Adapt then unlearn: Exploring parameter space semantics for unlearning in generative adversarial networks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=jAHEBivObO.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

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*, pp. 1755–1764, 2024.

Yimeng Zhang, Jinghan Jia, Xin Chen, Aochuan Chen, Yihua Zhang, Jiancheng Liu, Ke Ding, and Sijia Liu. To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images... for now. *arXiv preprint arXiv:2310.11868*, 2023.

A ALGORITHM

Algorithm 1 UOT-Unlearn for Class Unlearning

Require: Pretrained generator G_{pre}, generator G_θ initialized from G_{pre}, dual potential v_ϕ, and precomputed forget anchor μ_f

1: **for** each training iteration **do**
2: Sample three independent noise batches $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3 \sim \mu_Z$
3: Evaluate $c_{\mathrm{ul}}(G_{\mathrm{pre}}(x_0), G_\theta(x_0))$ for $x_0 \in \mathcal{B}_1 \cup \mathcal{B}_3$ using Eq. 16
4: **Update the dual potential**
5: $\mathcal{L}_v \leftarrow \frac{1}{|\mathcal{B}_1|} \sum_{x_0 \in \mathcal{B}_1} \Psi_1^*(v_\phi(G_\theta(x_0)) - c_{\mathrm{ul}}(G_{\mathrm{pre}}(x_0), G_\theta(x_0))) + \frac{1}{|\mathcal{B}_2|} \sum_{x_0 \in \mathcal{B}_2} \Psi_2^*(-v_\phi(G_{\mathrm{pre}}(x_0)))$
6: Update ϕ by minimizing $\mathcal{L}_v$
7: **Update the generator**
8: $\mathcal{L}_G \leftarrow \frac{1}{|\mathcal{B}_3|} \sum_{x_0 \in \mathcal{B}_3} [c_{\mathrm{ul}}(G_{\mathrm{pre}}(x_0), G_\theta(x_0)) - v_\phi(G_\theta(x_0))]$
9: Update θ by minimizing $\mathcal{L}_G$
10: **end for**

B PROOF OF PROPOSITION 3.1

B.1 SETTING AND ASSUMPTIONS

We write $\mu = p_{\mathrm{pre}}$ and $\nu = p_{\mathrm{data}}$, both in $\mathcal{P}(\mathcal{X})$ with $\mathcal{X} \subseteq \mathbb{R}^d$ Borel. Throughout we assume $f(y) \neq 0$ for all $y \in \mathcal{X}$ and $\mu_f \neq 0$, so that $d_{\cos}(f(y), \mu_f)$ is well defined, and we set

$$g(y) := \big(m - d_{\cos}(f(y), \mu_f)\big)_+, \ \mathcal{R}_f = \{g > 0\}, \tag{19}$$

where $\mathcal{R}_f$ is the active forget region of Eq. (17). The objective of Eq. (9) is

$$F_\lambda(\pi) := \int_{\mathcal{X} \times \mathcal{X}} c_{\mathrm{ul}} \, d\pi + D_{\Psi_1}(\pi_0 | \mu) + D_{\Psi_2}(\pi_1 | \nu), \tag{20}$$

for $\pi \in \mathcal{M}_+(\mathcal{X} \times \mathcal{X})$, with c_{ul} given by Eq. (18). For an entropy function ψ (convex, lower semicontinuous, $\psi \geq 0$, $\psi(1) = 0$) and the Lebesgue decomposition $\pi_1 = r\nu + \pi_1^\perp$, we use the standard definition $D_\Psi(\pi_1|\nu) = \int \psi(r) \, d\nu + \psi'(\infty) \, \pi_1^\perp(\mathcal{X})$ with $\psi'(\infty) = \lim_{t \to \infty} \psi(t)/t$ (Liero et al., 2018a). The recession constant $\psi'(\infty)$ is what penalizes mass placed outside $\mathrm{supp}(\nu)$.

We now state the assumptions underlying Proposition 3.1.

Assumption 1 (Semantic separation)**.** $\mathrm{supp}(\nu) = \mathcal{S}_f \sqcup \mathcal{S}_r$ *with* $\mathcal{S}_f, \mathcal{S}_r$ *disjoint Borel sets, and there exists* $\varepsilon > 0$ *with*

$$d_{\cos}\big(f(y), \mu_f\big) \leq m - \varepsilon \ \ (y \in \mathcal{S}_f), \qquad d_{\cos}\big(f(y), \mu_f\big) \geq m \ \ (y \in \mathcal{S}_r). \tag{21}$$

Equivalently $g \geq \varepsilon$ *on* $\mathcal{S}_f$ *and* $g = 0$ *on* $\mathcal{S}_r$*; in particular* $\mathcal{S}_f \subseteq \mathcal{R}_f$ *and* $\mathcal{S}_r \cap \mathcal{R}_f = \varnothing$.

Assumption 2 (Finite transport budget)**.** μ *and* ν *have finite second moments. With* $p_f := \nu(\mathcal{S}_f) \in (0, 1)$ *and* $\nu_r := \nu(\cdot \mid \mathcal{S}_r)$*, we set* $C_{\mathrm{tr}} := \tau \, W_2^2(\mu, \nu_r) < \infty$.

Assumption 3 (Divergences)**.** Ψ_1 *is an arbitrary admissible entropy function and* $\Psi_2 = D_{\mathrm{KL}}$*, i.e.* $\psi_2(t) = t \log t - t + 1$*, so* $\psi_2(0) = 1$ *and* $\psi_2'(\infty) = +\infty$.

Assumption 1 is the formal counterpart of the margin condition in Proposition 3.1: it requires the threshold m of Eq. (18) to be large enough to cover the forget support and small enough not to intersect the retain support. Assumption 2 guarantees that transporting the pretrained distribution onto the retain classes has finite cost, and Assumption 3 specifies the divergence setting used in our analysis: an admissible source divergence and the KL divergence for the target marginal.

Remark 1 (Conjugate convention)**.** *The convex conjugate of the normalized KL entropy* $\psi(t) = t \log t - t + 1$ *is* $\psi^*(s) = e^s - 1$*, not* e^s*. The exponential penalty* $\Psi_1^*(s) = e^s$ *used in the implementation corresponds to the unnormalized entropy* $\psi_1(t) = t \log t - t$*, for which* $\psi_1(1) = -1 < 0$*; the*

two differ by an additive constant that does not affect the inner maximization but does shift objective values. Since the bounds derived below using C_0 rely on the nonnegativity of $D_{\Psi_1} \geq 0$, we use the normalized convention throughout the analysis.

Finally, define the λ-independent constant

$$C_0 := C_{\mathrm{tr}} + \log \frac{1}{1-p_f} \;=\; \tau\, W_2^2(\mu, \nu_r) + \log \frac{1}{1-p_f}. \tag{22}$$

B.2 Formal Statement and Proof

Before addressing the main proposition, we begin with an elementary bound on the unlearning cost

Lemma 1 (Cost domination)**.** *For the cost of Eq. (18), we have $c_{\mathrm{ul}}(x, y) \geq \lambda\, g(y) \geq 0$ for every $(x, y) \in \mathcal{X} \times \mathcal{X}$. Consequently $c_{\mathrm{ul}} \geq \lambda\varepsilon$ on $\mathcal{X} \times \mathcal{S}_f$, and $c_{\mathrm{ul}}(x, y) = \tau\|x - y\|_2^2$ for every $y \in \mathcal{S}_r$.*

Proof. If $y \in \mathcal{R}_f$, then $c_{\mathrm{ul}}(x, y) = \lambda(m - d_{\cos}(f(y), \mu_f)) = \lambda g(y)$; if $y \notin \mathcal{R}_f$, then $g(y) = 0$ and $c_{\mathrm{ul}}(x, y) = \tau\|x - y\|_2^2 \geq 0$. The consequences follow from Assumption 1. □

The following is the formal version of Proposition 3.1. We state the result for δ-approximate minimizers. This includes exact minimizers as the special case $\delta = 0$ and, for $\delta > 0$, avoids the need to establish the existence of an exact minimizer.

Proposition 2 (Forget suppression and confinement to $\mathrm{supp}(\nu)$)**.** *Let Assumptions 1–3 hold, let $\delta \geq 0$, and let $\pi^\lambda \in \mathcal{M}_+(\mathcal{X} \times \mathcal{X})$ satisfy $F_\lambda(\pi^\lambda) \leq \inf F_\lambda + \delta$. Write π_1^λ for its target marginal and $M_\lambda := \pi_1^\lambda(\mathcal{X})$ for its total mass, and let $\phi := \psi_2|_{[0,1]}$, which is a strictly decreasing bijection of $[0, 1]$ onto itself. Then, for every $\lambda > 0$:*

(i) $\inf F_\lambda \leq C_0$*, and hence $F_\lambda(\pi^\lambda) \leq C_0 + \delta$.*

(ii) $\pi_1^\lambda \ll \nu$*; in particular π_1^λ is concentrated on $\mathcal{S}_f \cup \mathcal{S}_r$.*

(iii) $\pi_1^\lambda(\mathcal{S}_f) = \pi_1^\lambda(\mathcal{R}_f) \leq \dfrac{C_0 + \delta}{\varepsilon} \cdot \dfrac{1}{\lambda}$.

(iv) $D_{\Psi_2}(\pi_1^\lambda|\nu) \leq C_0 + \delta$ *and* $\psi_2(M_\lambda) \leq C_0 + \delta$*. If moreover $C_0 + \delta < 1$, then $M_\lambda \geq M_* := \phi^{-1}(C_0 + \delta) > 0$.*

(v) If $C_0 + \delta < 1$, the normalized target marginal $\bar{\pi}_1^\lambda := \pi_1^\lambda / M_\lambda \in \mathcal{P}(\mathcal{X})$ satisfies

$$\bar{\pi}_1^\lambda(\mathcal{S}_f) \leq \frac{C_0 + \delta}{M_*\, \varepsilon} \cdot \frac{1}{\lambda}, \quad \bar{\pi}_1^\lambda(\mathcal{S}_r) \geq 1 - \frac{C_0 + \delta}{M_*\, \varepsilon} \cdot \frac{1}{\lambda}. \tag{23}$$

Part (ii) states that unlearning does not leak probability mass into off-support, noise-like regions, and part (iii) is the quantitative erasure statement: the mass placed on the forget region decays as $O(1/\lambda)$. Since the UOT plan is not constrained to preserve total mass, parts (iv) and (v) convert these into a statement about the normalized marginal, which is the distribution that the unlearned generator is trained to produce; the smallness condition $C_0 + \delta < 1$ is what rules out a degenerate solution with vanishing mass. Taking $\delta = 0$ recovers Proposition 3.1.

Proof. **(i)** Let $\hat{\pi}$ be an optimal coupling for $W_2^2(\mu, \nu_r)$, which exists by Assumption 2, so that $\hat{\pi}_0 = \mu$ and $\hat{\pi}_1 = \nu_r$. Since ν_r is concentrated on $\mathcal{S}_r$, Lemma 1 gives $\int c_{\mathrm{ul}}\, d\hat{\pi} = \tau \int \|x - y\|_2^2\, d\hat{\pi} = C_{\mathrm{tr}}$; that is, the λ-dependent branch of the cost is never activated. Next, $\hat{\pi}_0 = \mu$ yields $D_{\Psi_1}(\hat{\pi}_0|\mu) = \psi_1(1) = 0$. Finally, $d\nu_r/d\nu = (1 - p_f)^{-1}\mathbf{1}_{\mathcal{S}_r}$, so

$$\begin{aligned} D_{\Psi_2}(\nu_r|\nu) &= (1 - p_f)\, \psi_2\Big(\tfrac{1}{1-p_f}\Big) + p_f\, \psi_2(0) \\ &= \Big[\log \tfrac{1}{1-p_f} - 1 + (1 - p_f)\Big] + p_f \\ &= \log \frac{1}{1 - p_f}. \end{aligned} \tag{24}$$

Summing the three terms gives $\inf F_\lambda \le F_\lambda(\hat{\pi}) = C_0$, uniformly in λ. Intuitively, $\log \frac{1}{1-p_f}$ is the divergence price of deleting the forget class from the target marginal, and it is this price that the forget cost $\lambda\varepsilon$ must exceed.

(ii) All three terms of equation 20 are nonnegative, so (i) gives $D_{\Psi_2}(\pi_1^\lambda|\nu) \le C_0 + \delta < \infty$. Writing $\pi_1^\lambda = r\nu + \pi_1^{\lambda,\perp}$, the definition of D_{Ψ_2} yields $\psi_2'(\infty)\,\pi_1^{\lambda,\perp}(\mathcal{X}) \le C_0 + \delta$, and $\psi_2'(\infty) = +\infty$ (Assumption 3) forces $\pi_1^{\lambda,\perp} = 0$. Hence $\pi_1^\lambda \ll \nu$ and π_1^λ is concentrated on $\mathrm{supp}(\nu) = \mathcal{S}_f \cup \mathcal{S}_r$.

(iii) Using $\int g(y)\, d\pi^\lambda(x,y) = \int g\, d\pi_1^\lambda$, Lemma 1 and the nonnegativity of the divergence terms give

$$C_0 + \delta \;\ge\; \int c_{\mathrm{ul}}\, d\pi^\lambda \;\ge\; \lambda \int_{\mathcal{S}_f} g\, d\pi_1^\lambda \;\ge\; \lambda\varepsilon\, \pi_1^\lambda(\mathcal{S}_f). \tag{25}$$

For the equality, part (ii) gives $\pi_1^\lambda(\mathcal{R}_f) = \pi_1^\lambda(\mathcal{R}_f \cap \mathcal{S}_f) + \pi_1^\lambda(\mathcal{R}_f \cap \mathcal{S}_r)$, while Assumption 1 gives $\mathcal{S}_f \subseteq \mathcal{R}_f$ and $\mathcal{R}_f \cap \mathcal{S}_r = \varnothing$.

(iv) The divergence bound follows from (i) and nonnegativity of the remaining terms. By (ii), $r = d\pi_1^\lambda/d\nu$ satisfies $\int r\, d\nu = M_\lambda$; since ν is a probability measure and ψ_2 is convex, Jensen's inequality gives

$$C_0 + \delta \;\ge\; \int \psi_2(r)\, d\nu \;\ge\; \psi_2\Big(\int r\, d\nu\Big) = \psi_2(M_\lambda). \tag{26}$$

On $[0,1]$, the function $\phi = \psi_2|_{[0,1]}$ is strictly decreasing from $\phi(0) = 1$ to $\phi(1) = 0$. Hence, if $M_\lambda \le 1$, the inequality $\psi_2(M_\lambda) \le C_0 + \delta < 1$ implies

$$M_\lambda \ge \phi^{-1}(C_0 + \delta).$$

If $M_\lambda > 1$, then $\phi^{-1}(C_0 + \delta) \le 1 < M_\lambda$. Therefore,

$$M_\lambda \ge M_* := \phi^{-1}(C_0 + \delta) > 0.$$

(v) Divide (iii) by M_λ and use $M_\lambda \ge M_*$ from (iv). By (ii) the normalized measure $\bar{\pi}_1^\lambda$ is concentrated on $\mathcal{S}_f \cup \mathcal{S}_r$, so $\bar{\pi}_1^\lambda(\mathcal{S}_f) + \bar{\pi}_1^\lambda(\mathcal{S}_r) = 1$, which gives the second inequality. □

C Class Unlearning: Implementation Details

C.1 Experimental Settings

In our unlearning experiments, we evaluate our approach on both unconditional and class-conditional generation tasks across different resolutions. To ensure reproducibility, all generative models, classifiers, and feature extractors used in our experiments are adopted from publicly available pretrained checkpoints.

- **CIFAR-10 (Krizhevsky et al., 2009) (Unconditional Generation):** We utilize Consistency Trajectory Model (CTM) (Kim et al., 2024) and Meanflow (Geng et al., 2025) models as unconditional generators. We focus on the class unlearning task—erasing a specific single class from the dataset—and conduct separate unlearning experiments targeting Class 1, Class 6, and Class 8.
- **ImageNet-256 (Russakovsky et al., 2015) (High-Resolution Class-Conditional Setting):** We employ the Meanflow (Geng et al., 2025) model (SiT-XL/2) and operate over the *full* set of 1,000 ImageNet classes. During the generator update, the conditioning labels of the retain batch are sampled uniformly from the 999 non-forget classes, and the forget label is additionally injected into each batch. To eliminate cherry-picking concerns, we evaluate **five forget classes**, obtained by randomly selecting five distinct subclasses and then randomly selecting one class from each subclass: Class 84 (peacock, *bird*), Class 207 (golden retriever, *mammal*), Class 402 (acoustic guitar, *musical instrument*), Class 555 (fire engine, *wheeled vehicle*), and Class 963 (pizza, *food*). For each forget class, a designated retain (drift) target c_r is selected as the top-1 nearest class under ResNet-50 cosine similarity, excluding the forget class itself; the class-conditional cost that uses c_r is detailed in Section C.3.

- **Forget Anchor (μ_f) Calculation:** To compute the unlearning cost c_{ul} and construct the forget anchor (centroid) μ_f in the feature space, we employ a pretrained feature extractor $f(\cdot)$ on a small subset of sampled images from the forget class. Specifically, we utilize 512 images for the CIFAR-10 dataset and 260 images for the ImageNet-256 dataset (10% for CIFAR-10 and 20% for ImageNet). This demonstrates that our proposed method requires minimal data to accurately identify and erase the target concept.
- **Strict Unlearning Scenario:** Furthermore, we operate under a strict unlearning scenario where retain data is inaccessible and only a single pretrained model checkpoint is available. This specific checkpoint constraint necessitates structural modifications to the baseline methods, which are detailed in Section D.

C.2 Pretrained Models and Evaluation Networks

Pretrained Generative Models For unconditional generation on the CIFAR-10 dataset, we adopt the pretrained CTM (FID: 1.73) and Meanflow (FID: 2.80) checkpoints. For high-resolution conditional generation on ImageNet-256, we utilize the pretrained conditional Meanflow checkpoint (SiT-XL/2) with an initial FID of 3.43.

Evaluation Networks To rigorously evaluate the efficacy of our unlearning framework, we employ the following pretrained evaluation networks:

- **Classifier (for PUL calculation):** For CIFAR-10, we use a pretrained DenseNet-121 model (Phan, 2021) which achieves an accuracy of 94.06%. For ImageNet-256, we employ a pretrained ViT-L/16 model (Wightman, 2019), achieving an accuracy of 88.55%.
- **FID Calculation:** We compute the Fréchet Inception Distance (FID) and unlearned-FID (u-FID) using the standard Inception-v3 network.

C.3 UOT-Unlearn Implementation

Network Architecture Our proposed UOT-Unlearn framework consists of a generator and a discriminator. Since the primary objective is to unlearn specific concepts from a pretrained model, the generator is directly initialized with the weights of the fully pretrained CTM or Meanflow checkpoint. For the discriminator, we adopt the exact architecture proposed in the standard Unbalanced Optimal Transport (UOT) generative model (Choi et al., 2023). Specifically, we use their small ResNet-based discriminator variant, featuring a base channel size of 64, LeakyReLU (0.2) activations, and a minibatch standard deviation layer.

C.4 Feature Extractor and Forget Anchor

To extract feature representations and compute the forget centroid, we utilize the penultimate layer of a pretrained ResNet-56 model (94.37% accuracy) (Chen, 2025) for CIFAR-10, and a pretrained ResNet-50 model (81.19% accuracy) (PyTorch Contributors, 2023) for ImageNet-256. To ensure a strictly fair and unbiased evaluation, we intentionally employ these ResNet-based feature extractors during the unlearning phase, which are distinctly separated from the classifiers (DenseNet-121 (Phan, 2021) and ViT-L/16 (Wightman, 2019)) used later for calculating the Probability of Unlearning (PUL) metric.

Training Hyperparameters For the CIFAR-10 unlearning experiments, we use a batch size of 128. Regarding the exponential moving average (EMA) of the generator weights, we disable EMA during the CTM unlearning process, whereas we set the EMA decay rate to 0.99 for the Meanflow unlearning process. For the high-resolution ImageNet-256 experiments, we utilize a batch size of 16 and set the EMA decay rate to 0.995. All other optimization settings strictly follow the default configuration provided in the original UOT implementation (e.g., Adam optimizer with learning rates of 1.6×10^{-4} for the generator and 1.0×10^{-4} for the discriminator, and $\tau = 10^{-4}$).

Conditional Extension for ImageNet-256 The cost function above, used for the unconditional CIFAR-10 experiments, is defined for an unconditional generator. For the class-conditional

ImageNet-256 backbone, the generator $G_\theta(x_0, y)$ additionally takes a class label y. We extend the cost with a retain-class conditional anchor for forget-labeled samples that have already escaped the active forget region $\mathcal{R}_f$. Let c_f be the forget label, c_r the designated retain target class, chosen as the top-1 nearest class under ResNet-50 cosine similarity, μ_f the forget anchor, and $d_{\cos}(x_0, y) = 1 - \cos(f(G_\theta(x_0, y)), \mu_f)$. The conditional unlearning cost is

$$c_{\mathrm{ul}}(x_0, y) = \begin{cases} \lambda \max\big(0,\ m - d_{\cos}(x_0, y)\big), & G_\theta(x_0, y) \in \mathcal{R}_f, \\ \tau \left\| G_\theta(x_0, y) - G_{\mathrm{pre}}(x_0, c_r) \right\|_2^2, & y = c_f,\ G_\theta(x_0, y) \notin \mathcal{R}_f, \\ \tau \left\| G_\theta(x_0, y) - G_{\mathrm{pre}}(x_0, y) \right\|_2^2, & \text{otherwise.} \end{cases} \tag{27}$$

The first case remains the margin-based forget term used in the unconditional objective, separating generated samples inside the forget region $\mathcal{R}_f$ from the forget anchor μ_f. The second case applies to forget-labeled samples after they have escaped $\mathcal{R}_f$, supplying a retain-class conditional anchor that guides them toward the neighboring retain class c_r. This makes explicit, in a label-conditioned generator, the probability redistribution that is handled implicitly in the unconditional setting. The third case is the standard retention term for the remaining samples, anchoring them to their corresponding pretrained outputs.

C.5 Computing Infrastructure

Hardware and software. CIFAR-10 experiments run on a single NVIDIA A100 80 GB GPU, while the high-resolution ImageNet-256 experiments use two NVIDIA A100 80 GB GPUs. The software stack is Ubuntu 24.04 LTS, Python 3.12, PyTorch 2.8 with CUDA 12.9 and cuDNN 9.1, torchvision 0.22, and timm 1.0.28. FID and u-FID are computed with an Inception-v3 based script, which follows the standard protocol of Heusel et al. (2017).

D Baseline Methods

To evaluate the effectiveness of our proposed framework, we compare it against several representative unlearning baselines. We consider a strict unlearning scenario in which only a single pretrained checkpoint is available and no real retain data are accessible. Forget-class data are used only when required by a baseline. When a baseline requires retain-like samples, we do not use real retain data; instead, we construct pseudo-retain samples by generating images from the frozen pretrained model and filtering out the forget concept using an auxiliary classifier. Thus, all baselines operate under the same available resources: the pretrained checkpoint, forget data when required, generated pseudo-data, and the auxiliary classifier.

D.1 Gradient Ascent (GA)

For the Gradient Ascent baseline, we directly maximize the original training loss on the entire training set of the target class, denoted as $\mathcal{D}_{\text{forget}}$. Since we apply this to the Consistency Trajectory Model (CTM) and Meanflow models, the GA loss is simply the negative of their respective training objectives:

$$\mathcal{L}_{\text{GA}}(\theta) = -\mathbb{E}_{x \sim \mathcal{D}_{\text{forget}}}[\mathcal{L}_{\text{train}}(\theta; x)] \tag{28}$$

where $\mathcal{L}_{\text{train}}$ denotes either the CTM or Meanflow loss.

For the class-conditional setting (i.e., ImageNet-256), the training objective inherently depends on both the image data and the class condition. Let y_f denote the target forget class label. The GA loss is naturally extended to maximize the expected conditional training loss over the forget data:

$$\mathcal{L}_{\text{GA}}(\theta) = -\mathbb{E}_{x \sim \mathcal{D}_{\text{forget}}}[\mathcal{L}_{\text{train}}(\theta; x, y_f)] \tag{29}$$

where $\mathcal{L}_{\text{train}}(\theta; x, y_f)$ represents the conditional training objective for a given image-label pair. This formulation ensures the model parameters are updated to diverge from the original conditional distribution associated with y_f.

D.2 VARIATIONAL DIFFUSION UNLEARNING (VDU) (PANDA ET AL., 2024)

The Variational Diffusion Unlearning (VDU) framework formulates the unlearning objective by integrating the negative training loss with a parameter penalty term to prevent catastrophic degradation. The general VDU loss is defined as:

$$\mathcal{L}_{\text{VDU}}(\theta) = -(1-\gamma)\mathbb{E}_{x\sim\mathcal{D}_{\text{forget}}}\left[\mathcal{L}_{\text{train}}(\theta; x)\right] + \gamma \sum_{i=1}^{d} \frac{(\theta_i - \mu_i^*)^2}{2\sigma_i^{*2}}. \tag{30}$$

where γ is the penalty weight, d is the total number of model parameters, and μ_i^* and σ_i^* represent the empirical mean and standard deviation of the i-th parameter, respectively. For our experiments, we set $\gamma = 0.005$ for CTM and $\gamma = 0.001$ for Meanflow.

To compute the parameter statistics (μ_i^* and σ_i^*), the original VDU method relies on multiple historical checkpoints saved during the initial pretraining phase. Since our setting provides only a single pretrained checkpoint, these trajectory-based statistics are unavailable. We therefore use a single-checkpoint adaptation of VDU: we set $\mu_i^* = \theta_{\text{pre},i}$ and use a shared constant σ^* for all parameters. Under this approximation, the parameter penalty reduces to an L_2 regularizer toward the pretrained weights θ_{pre}:

$$\sum_{i=1}^{d} \frac{(\theta_i - \mu_i^*)^2}{2\sigma_i^{*2}} = \frac{1}{2\sigma^{*2}} \left\|\theta - \theta_{\text{pre}}\right\|_2^2. \tag{31}$$

This adapted VDU preserves the main structure of the original objective, combining a forget loss with a parameter-stability penalty, while making it compatible with our single-checkpoint setting.

D.3 SELECTIVE AMNESIA (SA) (HENG & SOH, 2023)

Existing machine unlearning methods, such as Selective Amnesia (SA) and Saliency Unlearning (SalUn), are primarily designed for conditional generative models. To apply these methods to our unconditional generators, we first construct a common base unlearning objective ($\mathcal{L}_{\text{base}}$) that utilizes noise mapping and a pseudo-retain dataset.

Common Base Objective ($\mathcal{L}_{\text{base}}$) Since our unconditional generator lacks explicit class conditions to locate the target concept, we use a pretrained auxiliary classifier to filter the prior noise space. Let $\mathcal{Z}_{\text{forget}}$ denote the set of prior noise vectors $x_0 \sim \mathcal{N}(0, I)$ that the generator maps to images classified as the target concept. We introduce a *Noise Mapping Loss* to force the generator to output pure random noise ϵ for these specific latents:

$$\mathcal{L}_{\text{forget_noise}}(\theta) = \mathbb{E}_{x_0\sim\mathcal{Z}_{\text{forget}},\epsilon\sim\mathcal{N}(0,I)}\left[\left\|G_\theta(x_0) - \epsilon\right\|_2^2\right] \tag{32}$$

Simultaneously, we formulate a retain loss using a fixed pseudo-retain dataset, $\mathcal{D}_{\text{pr}}$. We construct this set offline by generating images with the original generator and filtering out the target concept using the auxiliary classifier. Similarly, we define the pseudo-forget dataset $\mathcal{D}_{\text{pf}}$ as the generated samples classified as the target concept by the auxiliary classifier. In other words, $\mathcal{D}_{\text{pr}}$ and $\mathcal{D}_{\text{pf}}$ are disjoint classifier-filtered subsets of samples generated from the frozen pretrained generator, corresponding to non-target and target concepts, respectively. We apply the standard training objective to these samples:

$$\mathcal{L}_{\text{retain}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_{\text{pr}}}\left[\mathcal{L}_{\text{train}}(\theta; x)\right] \tag{33}$$

The base unlearning objective is the weighted sum of these two terms: $\mathcal{L}_{\text{base}} = \alpha\mathcal{L}_{\text{forget_noise}} + \beta\mathcal{L}_{\text{retain}}$. We empirically tuned the hyperparameters via grid search, utilizing $\alpha \in \{0.05, 0.1\}$ and $\beta \in \{0.5, 5.0\}$.

SA Objective To constrain parameter deviation during unlearning, SA incorporates an Elastic Weight Consolidation (EWC) penalty into the base loss. This requires the computation of the Fisher Information Matrix (FIM), denoted as F. We compute the FIM using the entire generated pseudo-dataset, $\mathcal{D}_{\text{pseudo}} = \mathcal{D}_{\text{pr}} \cup \mathcal{D}_{\text{pf}}$. In practice, a diagonal approximation is adopted where each diagonal

element F_i corresponding to parameter θ_i is computed as $F_i = \mathbb{E}_{x\sim\mathcal{D}_{\text{pseudo}}}[(\nabla_{\theta_i}\mathcal{L}_{\text{train}}(\theta_{\text{pre}}; x))^2]$. The final SA objective is:

$$\mathcal{L}_{\text{SA}}(\theta) = \mathcal{L}_{\text{base}} + \frac{\lambda_{\text{SA}}}{2}\sum_i F_i(\theta_i - \theta_{\text{pre},i})^2 \tag{34}$$

where θ_{pre} represents the pretrained weights, and the penalty scale λ_{SA} is empirically set (e.g., 5.0 or 1000.0 based on the model scale).

D.4 Saliency Unlearning (SalUn) (Fan et al., 2024)

SalUn prevents catastrophic forgetting by restricting weight updates to only the most salient parameters associated with the target concept. Rather than modifying the loss function directly with a penalty, SalUn optimizes the same base objective ($\mathcal{L}_{\text{base}}$) defined in Section D.3, but applies a binary gradient mask $\mathbf{M}$ during the update step:

$$\mathbf{M} = \mathbb{I}\left(\left|\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{forget}}}\left[\mathcal{L}_{\text{train}}(\boldsymbol{\theta}; \mathbf{x})\right]\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{pre}}\right| \geq \gamma\right) \tag{35}$$

While the original SalUn paper recommends a sparsity ratio of 50% for multi-step diffusion models, we empirically found that unfreezing such a large proportion of weights severely disrupts the single-pass mapping of one-step models, leading to a collapse in generation quality. Thus, we adjust the threshold γ so that only the top 5% most salient parameters are updated, leaving the remaining 95% frozen. During optimization, we freeze the non-salient weights and update only the salient parameters selected by the mask:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\left(\mathbf{M}\odot\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{base}}\right) \tag{36}$$

where $\odot$ is the element-wise product and η is the learning rate.

D.5 Conditional Adaptation for ImageNet-256

The base objective above is designed for unconditional CIFAR-10 generators, where target concepts are identified through an auxiliary classifier because class labels are unavailable. On the class-conditional ImageNet-256 backbone, the forget label c_f is given explicitly, so each baseline instead specifies how forget-labeled generations are redirected. We summarize the per-method label-mapping strategy used in this setting:

- **UOT-Unlearn (Ours):** forget-labeled samples that leave the forget region $\mathcal{R}_f$ are matched to pretrained samples conditioned on the top-1 nearest retain class c_r (Eq. 27); c_r is chosen once per forget class by a ResNet-50 cosine-similarity scan over the remaining 999 non-forget classes.
- **Selective Amnesia (SA):** fixed surrogate mapping. Following the original SA formulation, the forget label is mapped to a designated surrogate class c_r (i.e., the surrogate $q(x \mid c_f)$ is set to the retain-target distribution), and forget-labeled generations are trained to match pretrained samples of c_r.
- **Saliency Unlearning (SalUn):** per-step random relabeling (the RL variant). At every step the forget label is replaced by a label drawn uniformly from the remaining classes ($c' \neq c_f$), and the update is restricted to the salient parameters selected by the mask. We update the top 50% most salient parameters, following the original SalUn default, as the explicit class label in the conditional backbone provides sufficient localization to tolerate a higher update ratio than the 5% used in our unconditional setting. The mask is regenerated for each forget class.
- **Gradient Ascent (GA):** *no* label mapping. GA directly maximizes the conditional training loss on the forget data, $\mathcal{L}_{\text{GA}}(\theta) = -\,\mathbb{E}_{x\sim\mathcal{D}_{\text{forget}}}[\mathcal{L}_{\text{train}}(\theta; x, c_f)]$.
- **Variational Diffusion Unlearning (VDU):** *no* label mapping. VDU uses the same single-checkpoint adaptation described in Section D: a negative conditional training loss on the forget data plus an L_2 parameter-stability penalty toward θ_{pre}.

Thus SA, SalUn, and Ours perform label-guided redirection—each with a different target-selection rule (fixed surrogate, uniform random, and nearest-class, respectively)—whereas GA and VDU suppress the forget class without any explicit remapping. This isolates the benefit of UOT-Unlearn's nearest-class semantic drift over both mapping-based and suppression-based baselines.

E T2I STYLE UNLEARNING: IMPLEMENTATION DETAILS

E.1 EXPERIMENTAL SETUP

We evaluate UOT-Unlearn on text-to-image (T2I) style unlearning, forgetting the "Van Gogh" style. This experiment use SDXL-Turbo (Sauer et al., 2024), a one-step distilled T2I diffusion model, at 512×512 resolution with a single denoising step, so the generator is a one-step G_θ and no sampling trajectory needs to be differentiated through. Training operates entirely in the latent space of the frozen VAE ($4 \times 64 \times 64$); the decoder is never invoked during training, and the VAE is loaded only to obtain the spatial downsampling factor. Since real training data for SDXL-Turbo are unavailable, we instantiate the reference marginal ν using samples from the frozen pretrained model rather than real images. Therefore, *no real image data are used at any stage of unlearning*; the only inputs are text prompts. Both text encoders are frozen and prompt embeddings are computed without classifier-free guidance, matching the SDXL-Turbo inference recipe.

E.2 NETWORK ARCHITECTURE

Generator. G_θ is initialized from G_{pre} and adapted with LoRA (Hu et al., 2022) of rank $r = 16$ and scale $\alpha = 16$ (dropout 0) injected into the attention projections `to_q`, `to_k`, `to_v`, `to_out.0`; all remaining UNet weights stay frozen. Gradient checkpointing and memory-efficient attention are enabled for the generator.

Potential. Following SDXL-Lightning (Lin et al., 2024), we build v_ϕ from the pretrained diffusion model itself rather than from an external vision backbone: such a backbone has already been trained on the target distribution. We copy the encoder and midblock of the pretrained SDXL-Turbo UNet as our backbone (`conv_in`, the timestep and SDXL added-condition embeddings, the down blocks and the mid block; the decoder half is discarded) and pass the resulting midblock features to a prediction head. The head consists of simple blocks of 4×4 convolution with a stride of 2, `GroupNorm`(32), and `SiLU`, after which the 1280-channel representation is globally pooled and projected to a single value. The backbone carries the *same* LoRA configuration as the generator, while the head is trained in full. Unlike Lin et al. (2024), the backbone is applied to a single input, and the output is left unbounded instead of being clamped to $[0, 1]$ by a sigmoid, since v_ϕ is a Kantorovich potential entering the objective through Ψ^* rather than a probability.

E.3 FEATURE EXTRACTOR AND FORGET ANCHOR

Teacher feature embedding. To compute the unlearning cost c_{ul}, forget centroid is measured in the teacher feature space. We employ the frozen teacher UNet as a feature extractor, denoted by $f(\cdot)$. We perturb the latent additively, $\tilde{x} = \hat{x} + \sigma\varepsilon$ with $\sigma = 0.1$, run the teacher at a fixed timestep $t = 100$, and capture activations at `down_blocks.2`, `mid_block` and `up_blocks.0`. Each activation is average-pooled with a 4×4 kernel of stride 4, flattened, and the resulting feature vectors are concatenated and finally ℓ_2-normalized.

Forget anchor. The forget anchor (centroid) μ_f is estimated once, before training, from the frozen reference model: for each prompt in the forget prompt list we generate 32 samples (in chunks of 8), embed them with frozen teacher UNet extractor $f(\cdot)$, and average over all prompts and samples, followed by a final ℓ_2 normalization. The centroid is then kept fixed throughout training.

E.4 T2I ADAPTATION FOR SDXL-TURBO

The base objective above is designed for the class-conditioned ImageNet-256 setting, where the forget concept is specified by a class label c_f. In the SDXL-Turbo text-to-image setting, conditioning is instead provided through text prompts. Accordingly, each baseline specifies how generations conditioned on forget prompts are modified. We summarize the objective for each baseline in text-to-image setting below.

- **UOT-Unlearn (Ours):** We adopt the same objective as described in Eq. 27, replacing class conditioning with prompt conditioning. Specifically, forget prompts (e.g., "A portrait of

Van Gogh") serve as the forget label c_f, while the corresponding retain prompts (e.g., "A portrait") serve as the retain label c_r in the T2I conditional objective.

- **Saliency Unlearning (SalUn):** We follow a baseline formulation described in Section D, adapted to the text-to-image setting. We first estimate a saliency map by accumulating $|\nabla_\theta \mathcal{L}|$ of a gradient-ascent objective over the forget prompts. We then retain the top 5% of entries using a single global threshold to form a binary mask m; The original work uses a 50% threshold. After applying the saliency map, we denote the unlearning model by θ_u. The SalUn objective in T2I setting is defined as follows:

$$\mathcal{L}_{\text{SalUn}}(\theta_u) = \left\|\epsilon_{\theta_u}(x_t, c_f) - \text{sg}(\epsilon_{\theta_u}(x_t, c_r))\right\|_2^2 + \alpha\left\|\epsilon_{\theta_u}(x_t, c_r) - \epsilon_{\text{pre}}(x_t, c_r)\right\|_2^2, \quad (37)$$

  where c_f and c_r denote the forget and retain prompts, respectively, and $\alpha = 0.1$ is the regularization parameter of the objective. To adapt SalUn as a one-step baseline, the loss is evaluated at the single denoising step of the SDXL-Turbo schedule. We fine-tune only the cross-attention projection layers (`attn2`) rather than the full UNet or LoRA, with AdamW at 1×10^{-5}, cosine-annealed to 1×10^{-6}, for 900 steps with 16 noise draws per prompt.

- **Gradient Ascent (GA):** *no* label mapping. GA directly maximizes the conditional training loss on forget prompts,

$$\mathcal{L}_{\text{GA}}(\theta) = -\,\mathbb{E}_{x\sim\mathcal{D}_{\text{forget}}}[\mathcal{L}_{\text{train}}(\theta; x, c_f)] \quad (38)$$

E.5 Optimization

Batch construction. The prompt dataloader yields one prompt per iteration, and that prompt is replicated 16 times against 16 independent noise draws. Every optimization step therefore estimates the training objective using 16 i.i.d. samples conditioned on a single prompt, keeping the transport cost comparable across the batch. Forget coverage is controlled at the dataset level: prompts appearing in both the training pool and the forget list are duplicated $3\times$, so that forget prompts are drawn roughly three times as often as their nominal share.

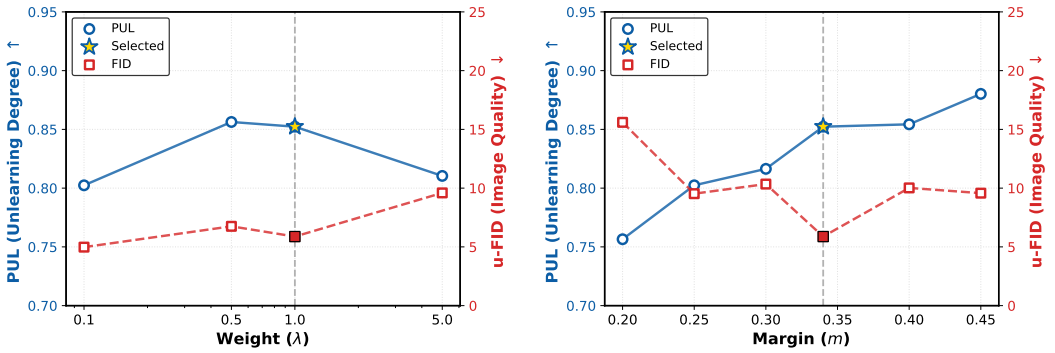
Hyperparameters. Generator and potential networks use Adam with $(\beta_1, \beta_2) = (0.5, 0.9)$, learning rates 1×10^{-4} (generator) and 2×10^{-4} (potential), cosine annealing to $\eta_{\min} = 10^{-5}$ over the full schedule, and gradient-norm clipping at 1.0. We use the KL divergence for both marginal penalties, i.e., $\Psi_1 = \Psi_2 = \text{KL}$. The retention weight is set to $\tau = 1 \times 10^{-4}$, the forget margin to $m = 0.6$, the forget weight to $\lambda = 15$, and the R_1 regularization coefficient to $\gamma = 5$. Training runs for 900 steps on a single GPU with seed 42.

F Ablation Study

F.1 Sensitivity to Key Hyperparameters

Through an ablation study on the CIFAR-10 CTM generator targeting Class 8, we investigate the sensitivity of the proposed UOT framework to its core hyperparameters: the forget loss weight λ and the distance margin m (Eq. (18) in the main text). The parameter λ controls the strength of the unlearning term. As illustrated in Figure 5 (left), setting $\lambda = 1.0$ provides the most favorable trade-off between concept erasure (PUL) and generative fidelity (u-FID). Relaxing this penalty ($\lambda = 0.1$) provides insufficient optimization signal for target removal. Conversely, an excessively large weight ($\lambda = 5.0$) destabilizes the unlearning process. Rather than monotonically improving erasure, this over-penalization alters the learned flow mapping, leading to a simultaneous degradation in both PUL and u-FID.

The margin m (Eq. (17) in the main text) defines the feature-space boundary required to accurately isolate and displace the target distribution. We observe an empirical optimum at $m = 0.34$, where the target concepts are successfully erased without corrupting adjacent semantic regions (Figure 5, right). A conservative margin ($m \leq 0.30$) fails to fully encapsulate the forget set. Consequently, the model generates residual target semantics that mismatch the true retain distribution, increasing the u-FID. On the other hand, an overly aggressive margin ($m \geq 0.40$) forces the unlearning objective to overlap with neighboring retain classes, which degrades the structural fidelity of non-target domains. This demonstrates that a precisely calibrated margin is essential for localized concept removal.



Sensitivity to Loss Weight (λ) Sensitivity to Margin (m)

Figure 5: **Ablation study of key hyperparameters.** Evaluated on the CIFAR-10 CTM generator, we independently vary the forget loss weight λ (left) and semantic distance margin m (right). The star ($\star$) marks the optimal configuration balancing PUL ($\uparrow$) and u-FID ($\downarrow$).

Method	Metric	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	Avg
VDU	PUL	16.81	53.86	46.10	13.73	60.51	36.42	51.25	38.65	48.43	32.29	39.81
	u-FID	4.82	71.98	40.77	4.61	24.87	46.70	47.11	53.81	52.07	88.21	43.50
WGAN	PUL	38.69	28.83	48.81	36.98	40.74	17.20	42.42	15.06	46.11	27.58	34.24
	u-FID	7.16	6.40	5.73	6.03	6.00	6.62	9.42	6.27	7.56	5.66	6.69
Ours	PUL	80.76	80.32	83.10	87.09	93.25	83.55	90.98	81.50	85.23	77.38	**84.32**
	u-FID	9.94	9.90	8.34	8.49	8.29	9.78	5.11	8.40	5.88	9.03	8.32

Table 4: **Comprehensive comparison across all CIFAR-10 classes (CTM).** The WGAN baseline applies a standard Wasserstein discriminator penalty using classifier-selected pseudo-retain samples $\mathcal{D}_{\text{pr}}$, without the UOT formulation or the unlearning cost c_{ul}. The substantial gap in PUL confirms that the combination of UOT and the unlearning cost is essential for effective concept erasure.

To obtain such a margin in a principled manner, building on the concepts introduced in Section 3.3 and Assumption A.1 in Supplementary Section A.1, we initialize the margin as follows. m is systematically initialized to the maximum cosine distance from the forget centroid. The furthest real or generated forget sample defines this initial semantic boundary.

F.2 Effect of the UOT Formulation

Beyond hyperparameter sensitivity, we examine whether the observed gains stem from the UOT formulation itself or merely from the adversarial training. We conduct a comprehensive evaluation across all ten CIFAR-10 classes using the CTM generator with a single set of hyperparameters. As a controlled baseline, we train a WGAN variant that applies a standard Wasserstein discriminator penalty between generated samples and a subset of generated samples identified as retain-like by the auxiliary classifier, without our unlearning cost c_{ul}. As summarized in Table 4, removing the UOT formulation causes the average PUL to collapse from 84.32% to 34.24%, while our full method maintains both a high PUL and a low u-FID across every class. This result confirms that the trade-off between concept erasure and generative fidelity is driven by the UOT framework, not by the discriminator alone.

F.3 Data Efficiency of the Forget Anchor

Moreover, we investigate the data efficiency of the forget anchor. As summarized in Table 5, our method demonstrates remarkable robustness to the number of anchor samples ($|\mathcal{D}_f|$ in Eq. (16) in the main text). For the target class, we varied the sample size from the full forget set (5,000 samples)

Number of Samples	PUL (%) ↑	u-FID ↓
64	85.03	7.80
128	85.03	8.08
256	**87.43**	7.65
512	86.83	**7.64**
1024	84.83	8.73
5000 (Full)	84.63	8.32

Table 5: **Data efficiency of the forget anchor.** Ablation on the number of anchor samples for the CIFAR-10 CTM generator. The method achieves robust concept erasure (PUL ↑) and generative quality (u-FID ↓) even with only 64 samples (vs. the full 5,000). Unlearning steps are fixed at 160.

	Linear: $\Psi_1^*(x) = x$		**Exponential:** $\Psi_1^*(x) = e^x$	
Unlearned Class	**PUL (%)** ↑	**u-FID** ↓	**PUL (%)** ↑	**u-FID** ↓
Class 1 (Auto)	75.35	13.12	**81.51**	**10.33**
Class 6 (Frog)	**91.94**	**5.29**	90.40	5.87
Class 8 (Ship)	81.84	7.81	**86.43**	**7.74**
Average	83.04	8.74	**86.11**	**7.98**

Table 6: **Impact of the penalty function.** Comparison between the linear identity function ($\Psi_1^*(x) = x$, exact matching) and our proposed exponential penalty ($\Psi_1^*(x) = e^x$, flexible matching) across different target classes on the CIFAR-10 CTM generator. While the exact matching constraint is effective on specific classes, the flexible exponential penalty consistently achieves a superior overall balance between concept erasure (PUL ↑) and generative fidelity (u-FID ↓). Best results are highlighted in **bold**.

Method	**C0**	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**Avg ± Std**
VDU	0.595	0.528	0.446	0.527	0.413	0.514	0.468	0.524	0.575	0.553	0.514 ± 0.054
WGAN	0.578	0.545	0.465	0.521	0.449	0.505	0.461	0.526	0.576	0.551	0.518 ± 0.044
Ours	0.567	0.529	0.479	0.524	0.466	0.517	0.460	0.527	0.579	0.553	0.520 ± 0.040

Table 7: **Membership inference AUROC across CIFAR-10 classes via PIA.** An average AUROC near 0.5 indicates that the forget concept cannot be reconstructed. All methods achieve a comparable defense level, but only ours simultaneously attains a high PUL.

down to 64. Notably, even with only 64 samples, the model maintains highly competitive forgetting efficacy (PUL) and generation quality (u-FID). This highlights that our framework achieves effective concept erasure without relying on extensive target data.

F.4 Impact of the Penalty Function

We investigate the impact of the penalty function Ψ_1^*, which controls the marginal relaxation of the source distribution as introduced in Section 2.2. In our framework, while the target distribution must remain flexible to enable concept erasure, the source distribution can be subjected to either an *exact* or a *flexible* matching condition. To evaluate this design choice, we compare our proposed exponential penalty ($\Psi_1^*(x) = e^x$, flexible matching) against a linear identity function ($\Psi_1^*(x) = x$, recovering exact OT matching) on the CIFAR-10 CTM generator.

As summarized in Table 6, enforcing exact source matching via the linear identity function yields inconsistent unlearning dynamics. Although it achieves effective concept erasure on specific domains (e.g., Class 6), this strict constraint struggles to maintain a stable trade-off across diverse classes. In contrast, relaxing this constraint with the exponential penalty allows for flexible source matching. This approach consistently achieves a superior overall balance between forgetting efficacy (PUL) and generative fidelity (u-FID). This demonstrates that allowing flexibility in the source distribution matching is crucial for robust and stable concept erasure.

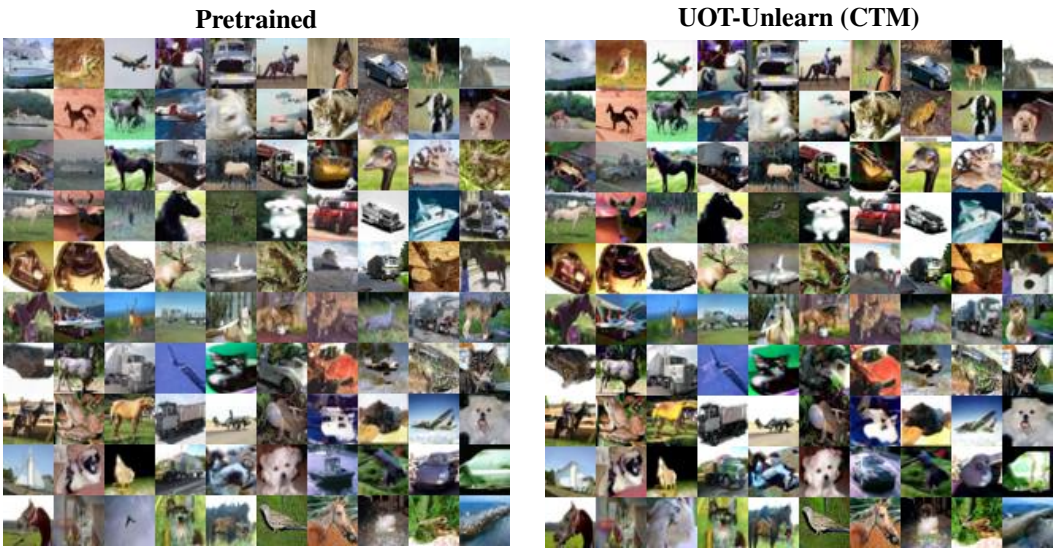


Figure 6: Visual comparison of images generated from the pretrained CTM versus our UOT-Unlearn-applied CTM trained to forget the Ship class (Class 8). The results demonstrate that UOT-Unlearn successfully erases the forget class, seamlessly transitioning it into features of other classes, while strictly preserving the visual quality and structural layout of the remaining retain classes.

F.5 Membership Inference Attack Evaluation

We also evaluate robustness against information recovery using the Proximal Initialization Attack (PIA) (Kong et al., 2024) with 1,000 samples per set. As reported in Table 7, all three methods yield an average AUROC close to 0.5, indicating that the forget concept cannot be reconstructed via membership inference. However, only our method distinctly excels in PUL, confirming we effectively unlearn concepts rather than merely destroying information.

G Visualization

G.1 CIFAR-10

We present visual results on the CIFAR-10 dataset based on the configurations that achieved the best performance in Table 2 of the main text. Figure 8 and Figure 9 compare the unlearning patterns of the forget class across the baselines and our UOT-Unlearn, using the CTM and MF models, respectively. Furthermore, Figure 6 and Figure 7 display how randomly generated images from the pretrained generators change after applying UOT-Unlearn. Overall, these visualizations demonstrate that our method not only erases the target concept but also naturally shifts its generation toward the retained classes, successfully preserving the structural layout of the original scenes.

G.2 ImageNet 256×256

As an extension to the results shown in Figure 2 of the main text, we provide additional visualization results on the ImageNet 256×256 dataset using the Meanflow model (Figure 10 and Figure 11). Using identical initial noises, we compare the pretrained model and UOT-Unlearn for a representative forget class (top row of each grid) against nine retain classes sampled from the remaining 999 classes. Visually, UOT-Unlearn effectively erases the specific features of the forget class while largely preserving the structural layouts of the retain classes. These qualitative observations align with our quantitative results, where UOT-Unlearn attains an average PUL of 98.75% and a u-FID of 8.54 across the five forget classes, close to the pretrained retain-quality reference (u-FID ≈ 5.52).

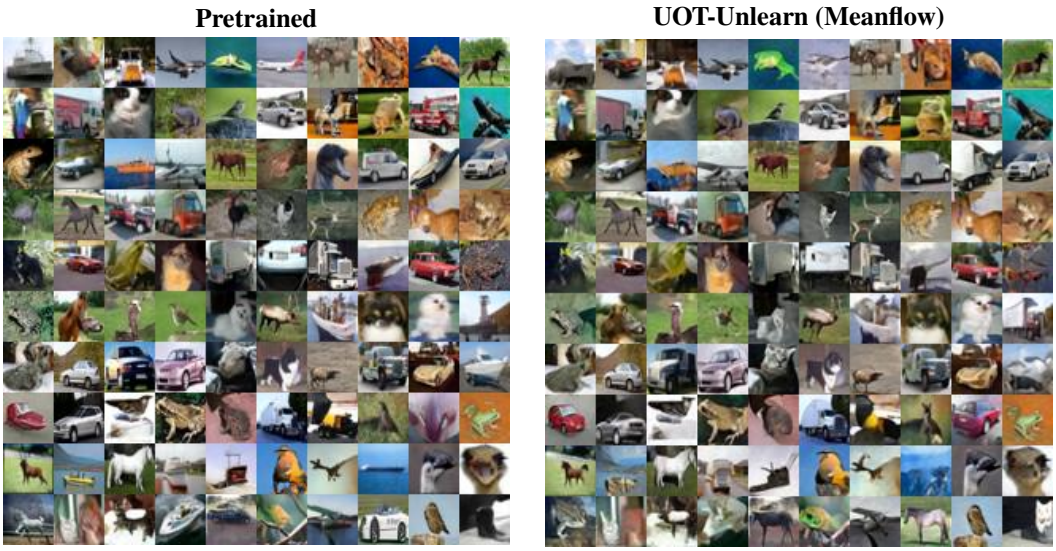


Figure 7: Visual comparison of randomly generated images from the Pretrained model and our UOT-Unlearn (MF) trained to forget the Ship class (Class 8), using the same initial noises.

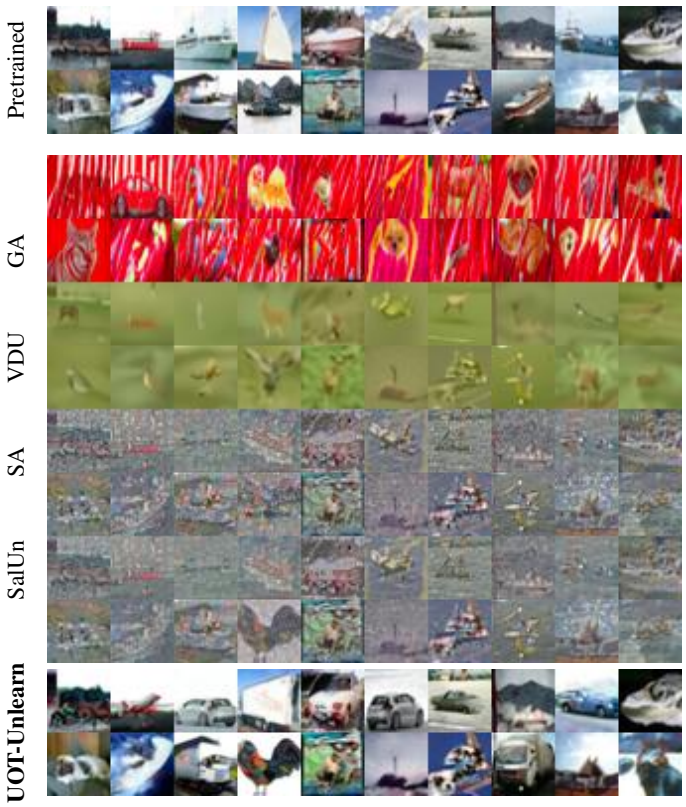


Figure 8: Qualitative comparison of unlearning methods on the Ship class (Class 8) using the Consistency Trajectory Model (CTM). Generated under fixed seed setting, this figure illustrates how the target class is erased across different approaches. Notably, unlike other baselines generate corrupted images, UOT-Unlearn demonstrates a distinct tendency to transition the forgotten concept into features of other classes while preserving the overall spatial layout.



Figure 9: Qualitative comparison of unlearning methods on the Ship class (Class 8) using the Meanflow (MF). Generated under fixed seed setting, this figure illustrates how the target class is erased across different approaches.



Pretrained



UOT-Unlearn (Meanflow)

Figure 10: Unlearning results on the ImageNet 256×256 dataset using the MeanFlow (MF) model. The top row of each generated batch displays the forget class (Peacock), while the subsequent bottom rows show 9 other random classes (retain classes) using the exact same initial noises.

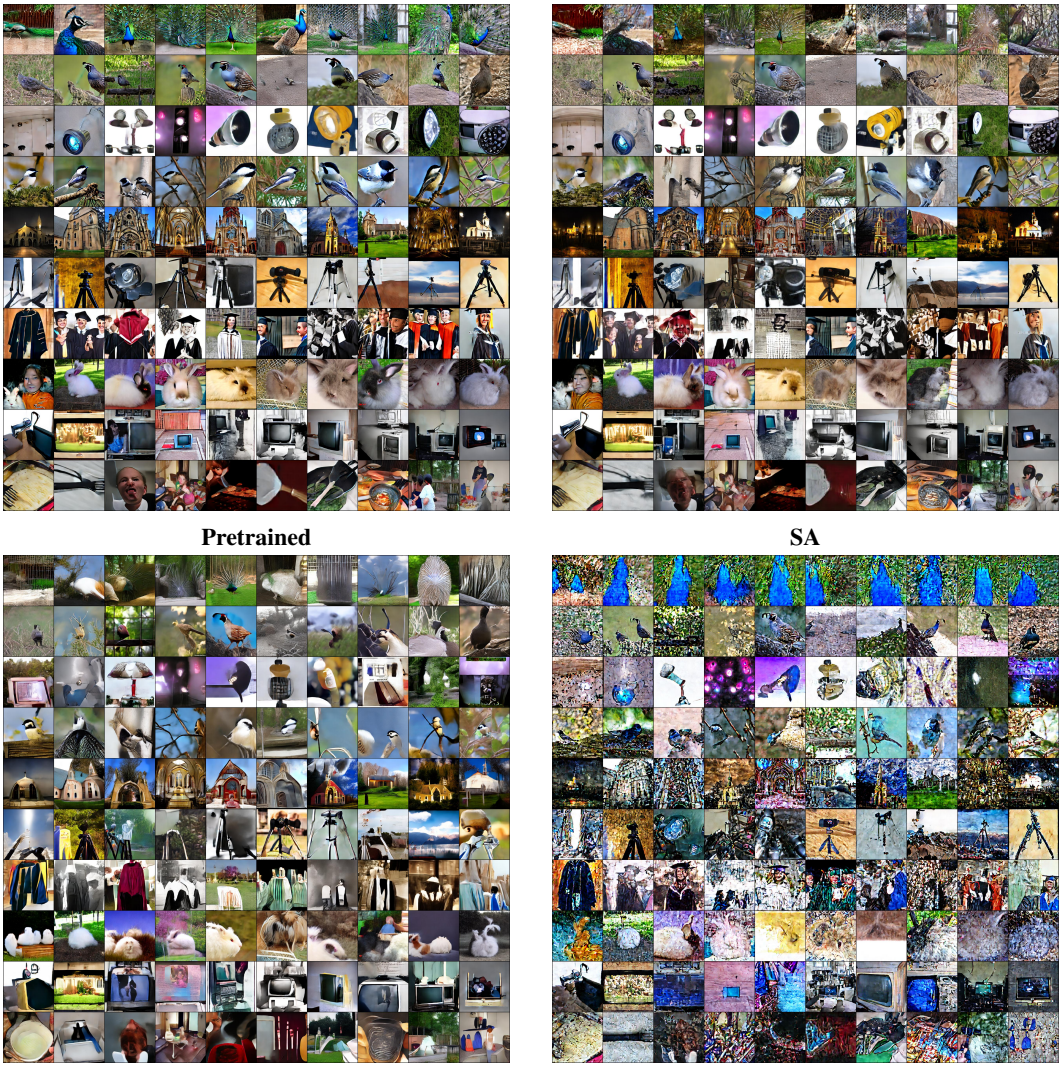


Figure 11: Qualitative comparison of baseline unlearning methods on ImageNet 256×256 using the Meanflow model. The top row of each generated batch displays the forget class (Peacock), while the subsequent bottom rows show 9 other random classes (retain classes) using the exact same initial noises.