Unlearning for One-Step Generative Models via Unbalanced Optimal Transport

Anonymous Authors[1]

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 ensuring the safety of these powerful generators remains entirely unexplored. Existing diffusion unlearning methods are inherently incompatible with these one-step models, as they rely on a multi-step iterative denoising process. In this work, we propose UOT-Unlearn, a novel plug-and-play 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 preserves overall generation fidelity via relaxed marginal constraints. By leveraging UOT, our method enables the probability mass of the forgotten class to be smoothly redistributed to the remaining classes, rather than collapsing into low-quality or noise-like samples. Experimental results on CIFAR-10 and ImageNet-256 demonstrate that our framework achieves superior unlearning success (PUL) and retention quality (u-FID), significantly outperforming baselines.

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 architectures. Previous techniques inherently rely on multi-step denoising processes, tweaking noise predictions or gradients at specific intermediate timesteps. In contrast, one-step models map noise to data in a single forward pass. Without intermediate steps during sampling, traditional step-by-step modifications are difficult to apply 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 designed specifically for one-step generative models. Our framework addresses the unlearning problem by exploiting 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 data distribution. In particular, we introduce an unlearning cost that penalizes samples belonging to the forget concept. This heavy penalization induces distribution

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

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 overall 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 benchmark datasets (e.g., CIFAR-10, ImageNet-256) using representative one-step architectures, such as CTM and Meanflow models, demonstrate that our method achieves superior class unlearning (PUL) and retention quality (u-FID).

2. Preliminaries

2.1. One-Step Generative Models 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 (Equation (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 seamlessly 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., 2018). 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:

$$D_{\Psi_1}(\pi_0|\mu) = \int_{\mathcal{X}} \Psi_1 \left(\frac{\mathrm{d}\pi_0(x)}{\mathrm{d}\mu(x)} \right) \mathrm{d}\mu(x), \tag{6}$$

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).

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:

$$\inf_{v_\phi} \left[\int_{\mathcal{X}} \Psi_1^* \left(-\inf_{T_\theta} [c(x, T_\theta(x)) - v_\phi(T_\theta(x))] \right) \mathrm{d}\mu(x) + \int_{\mathcal{Y}} \Psi_2^*(-v_\phi(y))\, \mathrm{d}\nu(y) \right], \tag{7}$$

where Ψ^* denotes the convex conjugate of Ψ. After training, the transport network T_θ learns an unbalanced optimal transport map between the source distribution μ and the target distribution ν.

In this work, we leverage the intrinsic trade-off in the UOT formulation between the transport cost and the marginal error to develop a novel unlearning algorithm. Intuitively, the UOT objective (Equation (5)) allows marginal mismatches when the resulting reduction in the transport cost outweighs the corresponding increase in the f-divergence penalties. This property is particularly suitable for machine unlearning. By designing a cost function that penalizes the generation of unlearn target, the UOT framework encourages the model to shift probability mass away from the *forget class* while maintaining the overall integrity of the remaining distribution through a relaxed matching process (Section 3.2).

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:

$$\mathbb{P}\big(G_\theta(x_0) \in \mathcal{S}_f\big) \to 0, \quad \text{and ideally,} \quad \mathbb{P}\big(G_\theta(x_0) \in \mathcal{S}_r\big) \to 1. \tag{8}$$

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

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 $c_{\mathrm{ul}}(\cdot,\cdot)$ 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., $\mu = p_{\mathrm{pre}}$, and the target distribution is the full data distribution $\nu = p_{\mathrm{data}}$.

$$\inf_{\pi \in \mathcal{M}_+(\mathcal{X}\times\mathcal{Y})} \left[\int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\, \mathrm{d}\pi(x,y) + D_{\Psi_1}(\pi_0|p_{\mathrm{pre}}) + D_{\Psi_2}(\pi_1|p_{\mathrm{data}}) \right], \tag{9}$$

In this formulation, the source distribution μ represents the starting point of the unlearning process. The transported marginal π_1 corresponds to the distribution produced by the updated generator after unlearning.

The UOT framework induces two desirable properties for π_1. First, the divergence term $D_{\Psi_2}(\pi_1|p_{\mathrm{data}})$ 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 optimal transported marginal $\pi_1^\star$

satisfies $\mathcal{S}_f \not\subset \mathrm{supp}(\pi_1^\star)$, which effectively suppresses the generation of the forget class and leads to

$$\mathbb{P}\big(G_\theta(x_0) \in \mathcal{S}_f\big) \to 0. \tag{10}$$

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), $D_{\Psi_2}(\pi_1|p_{\mathrm{data}})$ 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

$$\mathbb{P}\big(G_\theta(x_0) \in \mathcal{S}_r\big) \to 1. \tag{11}$$

Based on this formulation, we derive the unlearning objective using the semi-dual UOT framework. Let ΔT denote the unbalanced optimal transport map that transforms the pretrained distribution p_{pre} to the unlearned distribution π_1. Following the UOTM formulation (Equation (7)), we parameterize this transport map by a neural network ΔT_θ and obtain the following learning objective:

$$\inf_{v_\phi}\Bigg[\int_{\mathcal{X}} \Psi_1^*\Big(-\inf_{\Delta T_\theta}\big\{c(x_1,\Delta T_\theta(x_1)) - v_\phi(\Delta T_\theta(x_1))\big\}\Big)\mathrm{d}p_{\mathrm{pre}}(x_1) + \int_{\mathcal{Y}} \Psi_2^*(-v_\phi(y))\,\mathrm{d}\nu(y)\Bigg]. \tag{12}$$

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

$$\inf_{v_\phi}\Bigg[\int_{\mathcal{Z}} \Psi_1^*\Big(-\inf_{\Delta T_\theta}\big\{c(G_{\mathrm{pre}}(x_0),(\Delta T_\theta \circ G_{\mathrm{pre}})(x_0)) - v_\phi((\Delta T_\theta \circ G_{\mathrm{pre}})(x_0))\big\}\Big)\mathrm{d}\mu_Z(x_0) + \int_{\mathcal{Y}} \Psi_2^*(-v_\phi(y))\,\mathrm{d}\nu(y)\Bigg]. \tag{13}$$

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

$$\inf_{v_\phi}\Bigg[\int_{\mathcal{Z}} \Psi_1^*\Big(-\inf_{G_\theta}\big\{c_{\mathrm{ul}}(G_{\mathrm{pre}}(x_0),G_\theta(x_0)) - v_\phi(G_\theta(x_0))\big\}\Big)\mathrm{d}\mu_Z(x_0) + \int_{\mathcal{Y}} \Psi_2^*(-v_\phi(y))\mathrm{d}\nu(y)\Bigg]. \tag{14}$$

Finally, in our constrained unlearning setting where real retain data cannot be accessed, directly evaluating the expectation with respect to the data distribution $\nu = p_{\mathrm{data}}$ is infeasible. To address this issue, we approximate the target distribution using the pretrained distribution itself, i.e., $\nu \approx p_{\mathrm{pre}}$. This approximation enables a fully data-free optimization procedure while preserving the structural guidance provided by the original generator.

$$\inf_{v_\phi}\Bigg[\int_{\mathcal{Z}} \Psi_1^*\Big(-\inf_{G_\theta}\big\{c_{\mathrm{ul}}(G_{\mathrm{pre}}(x_0),G_\theta(x_0)) - v_\phi(G_\theta(x_0))\big\}\Big)\mathrm{d}\mu_Z(x_0) + \int_{\mathcal{Z}} \Psi_2^*(-v_\phi(G_{\mathrm{pre}}(x_0)))\mathrm{d}\mu_Z(x_0)\Bigg]. \tag{15}$$

3.3. Cost Design for Unlearning

To implement the objective in Equation (15), we introduce the unlearning cost function $c_{\mathrm{ul}}(\cdot,\cdot)$ 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 semantic center of the forget class in a feature space:

$$\mu_f = \frac{1}{|\mathcal{D}_f|}\sum_{x\in\mathcal{D}_f} f(x), \tag{16}$$

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 $\mathcal{S}_f$ (Equation (8)), 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:

$$\mathcal{R}_f = \Big\{x \in \mathcal{X} \;\Big|\; d_{\cos}\big(f(x),\mu_f\big) < m\Big\}. \tag{17}$$

where $d_{\cos}$ 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.

Based on this region, the unlearning cost is defined as

$$c_{\mathrm{ul}}(G_{\mathrm{pre}}(x_0),G_\theta(x_0)) = \begin{cases}\lambda\cdot\big(m - d_{\cos}(f(G_\theta(x_0)),\mu_f)\big), & \text{if } G_\theta(x_0)\in\mathcal{R}_f,\\ \tau\cdot\|G_{\mathrm{pre}}(x_0) - G_\theta(x_0)\|_2^2, & \text{otherwise.}\end{cases} \tag{18}$$

Minimizing this cost serves two primary roles: (i) For the forget cost, for generated samples inside $\mathcal{R}_f$, the hinge-like penalty pushes their features away from the anchor μ_f only until the distance exceeds the margin m. (i) For retain cost, for samples outside $\mathcal{R}_f$, treating the squared

Algorithm 1 Class Unlearning via UOT

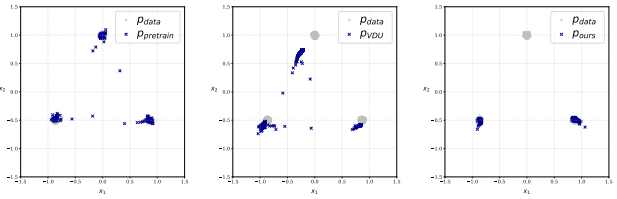
Require: pretrained generator G_{pre}, unlearned generator G_θ (initialized from G_{pre}), dual potential v_ϕ, precomputed forget anchor μ_f.

for each training iteration **do**
  Sample independent noise batches $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3 \sim \mu_Z$
  Compute $c_{\text{ul}}(G_{\text{pre}}(x_0), G_\theta(x_0))$ for $x_0 \in \mathcal{B}_1 \cup \mathcal{B}_3$ via Equation (18)
  Update Dual Potential
  $$\mathcal{L}_v \leftarrow \frac{1}{|\mathcal{B}_1|} \sum_{x_0 \in \mathcal{B}_1} \Psi_1^*\Big(v_\phi(G_\theta(x_0)) - c_{\text{ul}}(G_{\text{pre}}(x_0), G_\theta(x_0))\Big) + \frac{1}{|\mathcal{B}_2|} \sum_{x_0 \in \mathcal{B}_2} \Psi_2^*\Big(-v_\phi(G_{\text{pre}}(x_0))\Big)$$
  Update ϕ to minimize $\mathcal{L}_v$
  Update Generator
  $$\mathcal{L}_G \leftarrow \frac{1}{|\mathcal{B}_3|} \sum_{x_0 \in \mathcal{B}_3} \Big[c_{\text{ul}}(G_{\text{pre}}(x_0), G_\theta(x_0)) - v_\phi(G_\theta(x_0))\Big]$$
  Update θ to minimize $\mathcal{L}_G$
end for

L_2 distance as a transport cost preserves pixel-level fidelity by tightly anchoring the remaining classes to their original outputs, while naturally guiding expelled forget samples toward the nearest remaining classes. λ and τ act as balancing weights for the active forgetting and the fidelity retention, respectively. Through this design, our UOT framework safely redistributes the removed concepts into retain samples.

4. Related Works

Existing unlearning methodologies for generative models are often tailored to the iterative denoising structure of diffusion models. As a foundational baseline, Gradient Ascent (GA) (Thudi et al., 2022) attempts to reverse the training process by directly applying gradient ascent on the forgetting data. However, this naive parameter-space update often leads to severe instability and catastrophic forgetting, degrading the overall generation quality. To mitigate such issues, Selective Amnesia (SA) (Heng & Soh, 2023) uses Elastic Weight Consolidation (EWC) to penalize parameter changes based on the Fisher Information Matrix (FIM), but its reliance on expensive FIM calculations and generative replay limits real-time efficiency. Saliency Unlearning (SalUn) (Fan et al., 2024) identifies sensitive weights through gradient-based saliency to create forgetting masks, though its performance is highly sensitive to gradient and architectural biases. Variational Diffusion Unlearning (VDU) (Panda et al., 2026) introduces a variational inference framework that balances plasticity and stability to forget specific classes, reducing computational overhead but still tied to the noise scheduling and transitional kernels of diffusion models. See Appendix for further details.



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

Figure 1. **Unlearning results on a 2D toy dataset** where the forget mode is located 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.

Unlike these diffusion-centric approaches, our UOT-based framework (Section 3) 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 provides a systematic evaluation of our UOT-based unlearning framework. Our experiments are organized to comprehensively validate the core properties of the proposed method. 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 conduct an ablation study to characterize the sensitivity of the optimization dynamics to core hyperparameters. Detailed implementation settings, including specific network architectures and hyperparameter configurations, are provided in Appendix.

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). Because these methods are designed for multi-step iterative denoising, we adapt their objectives to operate within a single forward-pass generation framework. See Appendix for adaptation details.

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.

Table 1. **Comparison of data requirements for unlearning.** Unlike baseline methods that continuously require real data during training, our framework requires real forget data only **once** to pre-compute the forget centroid, relying solely on synthetic samples thereafter.

Method	Data Requirements		
	Real Forget	**Real Retain**	**Generated data**
Gradient Ascent (GA)	Required	None	None
VDU	Required	None	None
Selective Amnesia	None	None	Required
SalUn	Required	Required	Required*
UOT-Unlearn (Ours)	**Once**†	**None**	**Required**

* SalUn utilizes generated data only in specific large-scale generation tasks.
† Real forget data is used only once to pre-compute the forget centroid μ_f (Equation (16)) prior to unlearning.

Table 2. **Unlearning performance on CIFAR-10.** Best results are highlighted in **bold**, and second-best results are underlined. UOT-Unlearn consistently achieves the highest Percentage of Unlearning (PUL) while preserving the original FID.

Unlearned	Original	GA		SA		SalUn		VDU		UOT-Unlearn (Ours)	
Class	**FID** ↓	PUL (%) ↑	u-FID ↓	PUL (%) ↑	u-FID ↓	PUL (%) ↑	u-FID ↓	PUL (%) ↑	u-FID ↓	PUL (%) ↑	u-FID ↓
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 (Heng & Soh, 2023)											
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**

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 (Section 3.2). In contrast, while the baseline method (VDU (Panda et al., 2026)) is successful in unlearning the forget mode (at $(0, 1)$), the removed probability mass is redistributed into the invalid regions outside the support of p_{data}.

5.2. Image Unlearning Benchmarks

Experimental Setup. We evaluate our method on image-generation benchmarks. Our primary experiments are conducted on the CIFAR-10 dataset using Consistency Trajectory Models (CTM) (Kim et al., 2024) and MeanFlow (MF) (Geng et al., 2025) as representative one-step generative architectures. To assess concept erasure performance, we perform single-class unlearning targeting classes 1 (*automobile*), 6 (*frog*), and 8 (*ship*). We further scale UOT-Unlearn to ImageNet-256 using the class-conditional Meanflow model. To apply our unconditional unlearning framework, we marginalize over class labels, effectively treating it as an unconditional generator. We focus on aquatic classes to evaluate semantic shift. We use pretrained classifiers as the feature extractor $f(\cdot)$ for computing the unlearning cost c_{ul}. To construct the forget anchor μ_f, we use only 512 forget-class samples, demonstrating that our method requires minimal data.

Evaluation Metrics. 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. Let N_{pre} and N_{unl} denote the number of images generated by the pretrained and unlearned models, respectively, that are classified as the forget class. The PUL score is defined as

$$\text{PUL} = \frac{N_{\text{pre}} - N_{\text{unl}}}{N_{\text{pre}}} \times 100\%. \tag{19}$$

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

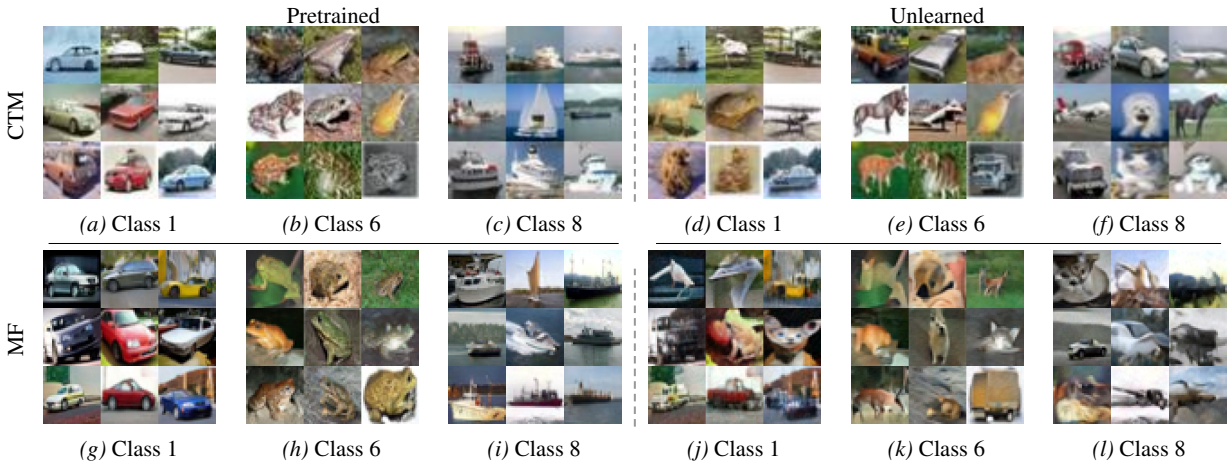


Figure 2. **Qualitative unlearning results on CIFAR-10**. Unconditional samples of target classes (1, 6, and 8) generated by CTM (top) and MF (bottom) architectures. To clearly illustrate the semantic erasure of targeted concepts, the *Unlearned* outputs are generated using the exact same initial noise seeds as their *Pretrained* counterparts.

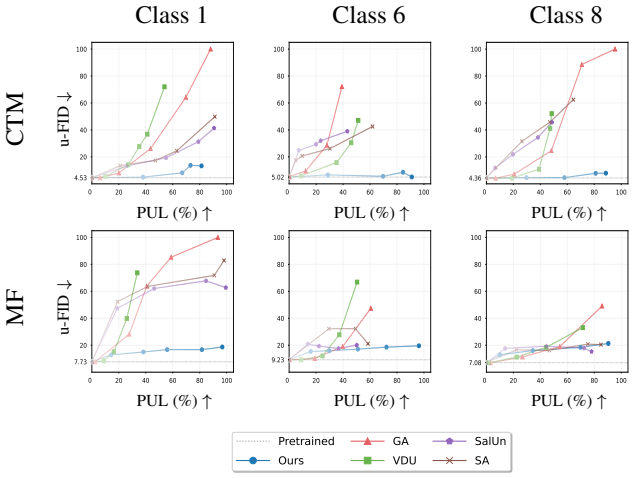


Figure 3. **Step-wise unlearning trajectories on CIFAR-10.** As unlearning progresses, concept erasure (PUL ↑) generally increases alongside worsened generative fidelity (u-FID ↓). Thus, step-wise evaluation is essential to assess this dynamic trade-off across unconditional CTM (top row) and MF (bottom row) models. Our framework consistently achieves a superior balance.

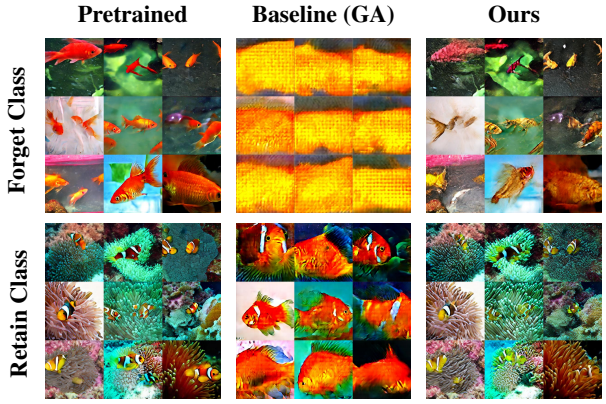


Figure 4. **Qualitative unlearning results on ImageNet-256 (Goldfish).** Generated samples for the *Forget* and *Retain* classes comparing the Pretrained model, Baseline (GA), and Ours.

retain images (the full training set excluding the forget class) and 45,000 images randomly generated by the unlearned model. On ImageNet-256, u-FID is computed against a localized retain subset of aquatic classes to precisely quantify unintended semantic degradation.

Experimental Results. As summarized in Table 2, our method achieves consistently high PUL scores across all targeted classes while maintaining an exceptionally low u-FID. Furthermore, Figure 3 highlights the step-wise unlearning dynamics. Typically, as training progresses, an increase in PUL often comes at the cost of a worsened u-FID (Gandikota et al., 2023; Heng & Soh, 2023; Fan et al., 2024). In contrast, our method demonstrates a remarkably favorable trajectory, effectively mitigating this common trade-off. Finally, 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 (see Appendix for other baselines).

High-resolution class-conditional generation on ImageNet-256 exposes the limitations of standard unlearning objectives. We focus our experimental setting on a localized subset of 37 semantically related aquatic classes, including the 'Goldfish' target. As reported in Table 3, our method achieves highly competitive concept erasure (PUL) while significantly outperforming baselines in generative fidelity (u-FID). Qualitatively, as shown in Figure 4, Gradient Ascent (GA) severely degrades generative quality, yielding corrupted patterns for both the forget and retain classes (see Appendix for other baselines). In contrast, our method redistributes forget class probability mass to semantically similar aquatic classes while maintaining retain distribution fidelity.

Table 3. **Unlearning performance on ImageNet-256 (Meanflow).** Best and second-best results are **bolded** and underlined, respectively. UOT-Unlearn achieves competitive PUL and significantly outperforms all baselines in u-FID. The original pretrained Meanflow model yields an FID of 11.57 on the retain classes.

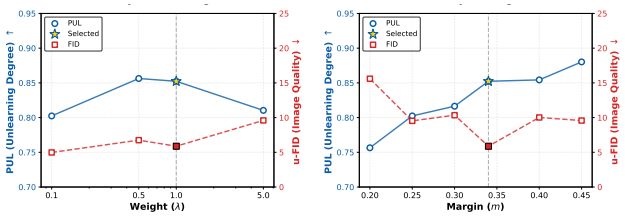
Method	PUL (%) ↑	u-FID ↓
GA (Gradient Ascent)	72.82	79.89
SA (Selective Amnesia)	76.66	42.37
SalUn	72.44	47.52
VDU	**85.32**	30.26
Ours (UOT-Unlearn)	85.08	**20.16**

5.3. Ablation Study

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 (Equation (18)). 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 (Equation (17)) 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.

Furthermore, we investigate the data efficiency of the forget anchor. As summarized in Table 4, our method demonstrates remarkable robustness to the number of anchor samples ($|\mathcal{D}_f|$ in Equation (16)). For the target class, we varied the sample size from the full forget set (5,000 samples) 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. An additional ablation study on the impact of the penalty function is provided in the Appendix.



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 (↑) and u-FID (↓).

Table 4. **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.

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

6. Conclusion

We introduced a machine unlearning framework for one-step generative models. Rather than adapting iterative denoising-based strategies, we cast concept erasure as an unbalanced optimal transport (UOT) problem and incorporate a transport cost directly into the single-pass mapping. This formulation redistributes probability mass away from the forget region without requiring access to real retain data.

Empirical results show that baselines adapted to the one-step regime consistently exhibit an efficacy–fidelity trade-off. Strong suppression often leads to significant deviations from the original data distribution, whereas milder interventions leave residual target semantics. In contrast, our method consistently attains high unlearning performance (PUL) while maintaining limited deviation in u-FID. These observations indicate that the UOT regularization induces a structured redistribution of probability mass rather than large-scale distortion of the learned distribution.

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.

References

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

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

Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., and Papernot, N. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021. doi: 10.1109/SP40001.2021.00019.

Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F.-X. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11): 3090–3123, 2018.

Choi, J., Choi, J., and Kang, M. 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.

Fan, C., Liu, J., Zhang, Y., Wei, D., Wong, E., and Liu, S. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, 2024.

Gandikota, R., Materzynska, J., Fiotto-Kaufman, J., and Bau, D. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2426–2436, 2023.

Geng, Z., Deng, M., Bai, X., Kolter, J. Z., and He, K. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025.

Heng, A. and Soh, H. 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.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. 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.

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

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

Kim, D., Lai, C.-H., Liao, W.-H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., and Ermon, S. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *International Conference on Learning Representations*, 2024.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kumari, N., Zhang, B., Wang, S.-Y., Shechtman, E., Zhang, R., and Zhu, J.-Y. Ablating concepts in text-to-image diffusion models. pp. 22634–22645, 10 2023. doi: 10.1109/ICCV51070.2023.02074.

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

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

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

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

Nguyen, T. T., Huynh, T. T., Ren, Z., Nguyen, P. L., Liew, A. W.-C., Yin, H., and Nguyen, Q. V. H. A survey of machine unlearning. *ACM Transactions on Intelligent Systems and Technology*, 16(5), 2025. doi: 10.1145/3749987.

Panda, S., Varun, M., Jain, S., Maharana, S. K., and Prathosh, A. Variational diffusion unlearning: A variational inference framework for unlearning in diffusion models. In *NeurIPS Safe Generative AI Workshop*, 2024.

Panda, S., Varun, M., Jain, S., Maharana, S. K., and Prathosh, A. Unlearning in diffusion models under data constraints: A variational inference approach. *Transactions on Machine Learning Research*, 2026.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Santambrogio, F. Optimal transport for applied mathematicians. 2015.

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

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. In *International Conference on Machine Learning*, 2023.

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

Tiwary, P., Guha, A., Panda, S., and AP, P. 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.

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

Zhang, G., Wang, K., Xu, X., Wang, Z., and Shi, H. 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.

A. Implementation Details

A.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. Rather than unlearning across all 1,000 classes, we restrict the conditioning labels during the generator update to a localized subset of 37 aquatic classes. Specifically, we choose "goldfish" as our target forget class, while the remaining 36 classes serve as semantically adjacent concepts. This localized constraint is intentionally designed to encourage the model to remap the forgotten concept into relevant domains, facilitating a smooth semantic shift rather than introducing unintended distributional shifts.

- **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 (representing exactly 20% of the approximately 1,300 available training images per class). 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 the original retain data is completely inaccessible, and only the fully pretrained checkpoint is available. This specific checkpoint constraint necessitates structural modifications to the baseline methods, which are detailed in Section B.

A.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. Following the ECCV anonymity policy, exact links to the pretrained models are omitted and will be added later.

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 which achieves an accuracy of 94.06%. For ImageNet-256, we employ a pretrained ViT-L/16 model, 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.

A.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.

Feature Extractor (for Centroid Calculation) To extract feature representations and compute the forget centroid, we utilize the penultimate layer of a pretrained ResNet-56 model (94.37% accuracy) for CIFAR-10, and a pretrained ResNet-50 model (81.19% accuracy) 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 and ViT-L/16) 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 8 and set the EMA decay rate to 0.999. 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 $\tau = 10^{-4}$).

B. Baseline Methods

To evaluate the effectiveness of our proposed framework, we compare it against several representative unlearning baselines. For fair comparison under our strict unlearning scenario (where only a single pretrained checkpoint is available and retain data is inaccessible), we establish the following baseline setups.

B.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:

$$\mathcal{L}_{GA}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_{\text{forget}}}[\mathcal{L}_{\text{train}}(\theta; x)] \tag{20}$$

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:

$$\mathcal{L}_{GA}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_{\text{forget}}}[\mathcal{L}_{\text{train}}(\theta; x, y_f)] \tag{21}$$

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.

B.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:

$$\mathcal{L}_{VDU}(\theta) = -(1-\gamma)\mathcal{L}_{\text{train}}(\theta; \mathcal{D}_{\text{forget}}) + \gamma \sum_{i=1}^{d} \frac{(\theta_i - \mu_i^*)^2}{2\sigma_i^{*2}} \tag{22}$$

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. However, under our strict setting where only a single fully pretrained checkpoint is provided, we adopt a practical alternative strategy to estimate these statistics. Starting from the pretrained checkpoint, we fine-tune the model on the full training dataset for 4 epochs (with a learning rate of 1×10^{-6} and a batch size of 128). We save a checkpoint at the end of every epoch. Combined with the initial pretrained checkpoint, this yields a total of 5 checkpoints (from 0 to 4 epochs), which we use to compute the required empirical mean and variance for the penalty term.

B.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:

$$\mathcal{L}_{\text{forget\_noise}}(\theta) = \mathbb{E}_{x_0 \sim \mathcal{Z}_{\text{forget}}, \epsilon \sim \mathcal{N}(0,I)} \left[\|G_\theta(x_0) - \epsilon\|_2^2 \right] \tag{23}$$

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. We apply the standard training objective to these samples:

$$\mathcal{L}_{\text{retain}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_{\text{pr}}} \left[\mathcal{L}_{\text{train}}(\theta; x) \right] \tag{24}$$

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:

$$\mathcal{L}_{\text{SA}}(\theta) = \mathcal{L}_{\text{base}} + \frac{\lambda_{\text{SA}}}{2} \sum_i F_i (\theta_i - \theta_{\text{pre},i})^2 \tag{25}$$

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).

B.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 B.3, but applies a binary gradient mask $\mathbf{M}$ during the update step:

$$\mathbf{M} = \mathbb{I}\left(\left| \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{\text{forget}}} \left[\mathcal{L}_{\text{train}}(\boldsymbol{\theta}; \mathbf{x}) \right] \big|_{\boldsymbol{\theta} = \boldsymbol{\theta}_{pre}} \right| \geq \gamma \right) \tag{26}$$

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 strictly adjust the threshold γ to enforce a sparsity ratio of 95%, selecting only the top 5% of the parameters to be updated. During optimization, we freeze the non-salient weights and update only the salient parameters selected by the mask:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \left(\mathbf{M} \odot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{base}} \right) \tag{27}$$

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

C. Ablation Study

C.1. 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

Table 5. **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**.

	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**

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 5, 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.

D. Visualization

D.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 6 and Figure 8 compare the unlearning patterns of the forget class across the baselines and our UOT-Unlearn, using the CTM and MF models, respectively. Furthermore, Figure 7 and Figure 9 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.

D.2. ImageNet 256×256

As an extension to the results shown in Fig. 4 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 across the target forget class (goldfish, top row) and 9 aquatic retain 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 robust metrics (85.08% PUL; u-FID 20.16 vs. pretrained FID 11.57).

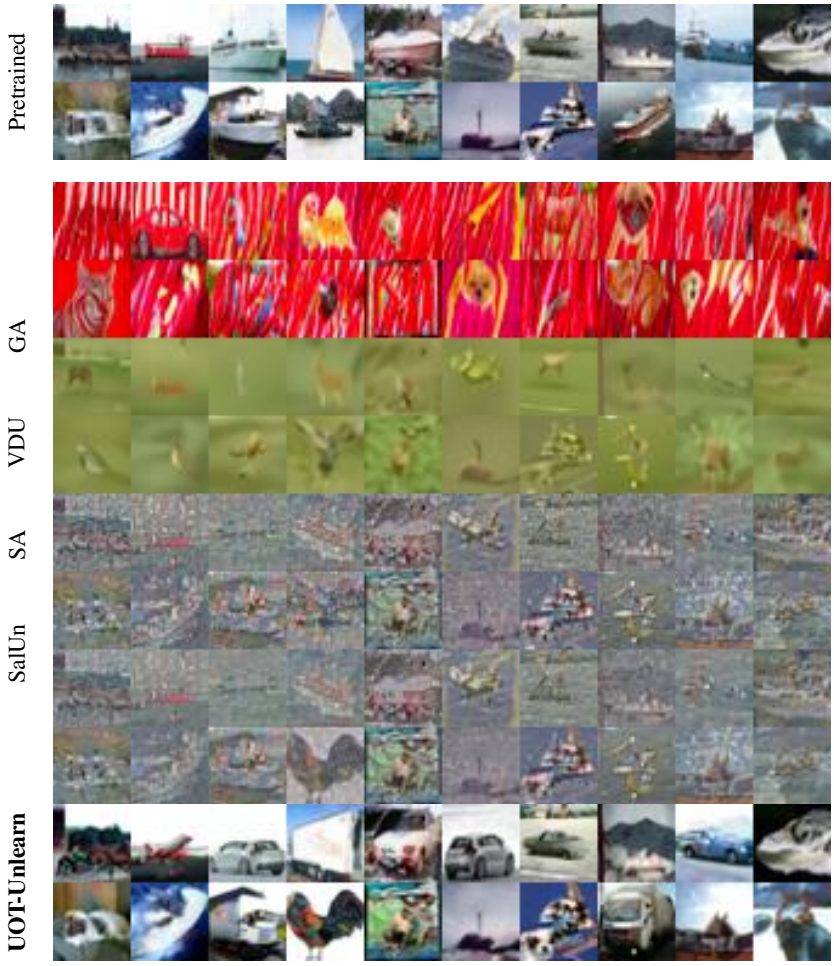


Figure 6. Qualitative comparison of unlearning methods on the Ship class (Class 8) using the Consistency Trajectory Model (CTM). Generated under fixed seed setting, this figure illustrates how the target class is erased across different approaches. Notably, unlike other baselines generate corrupted images, UOT-Unlearn demonstrates a distinct tendency to transition the forgotten concept into features of other classes while preserving the overall spatial layout.

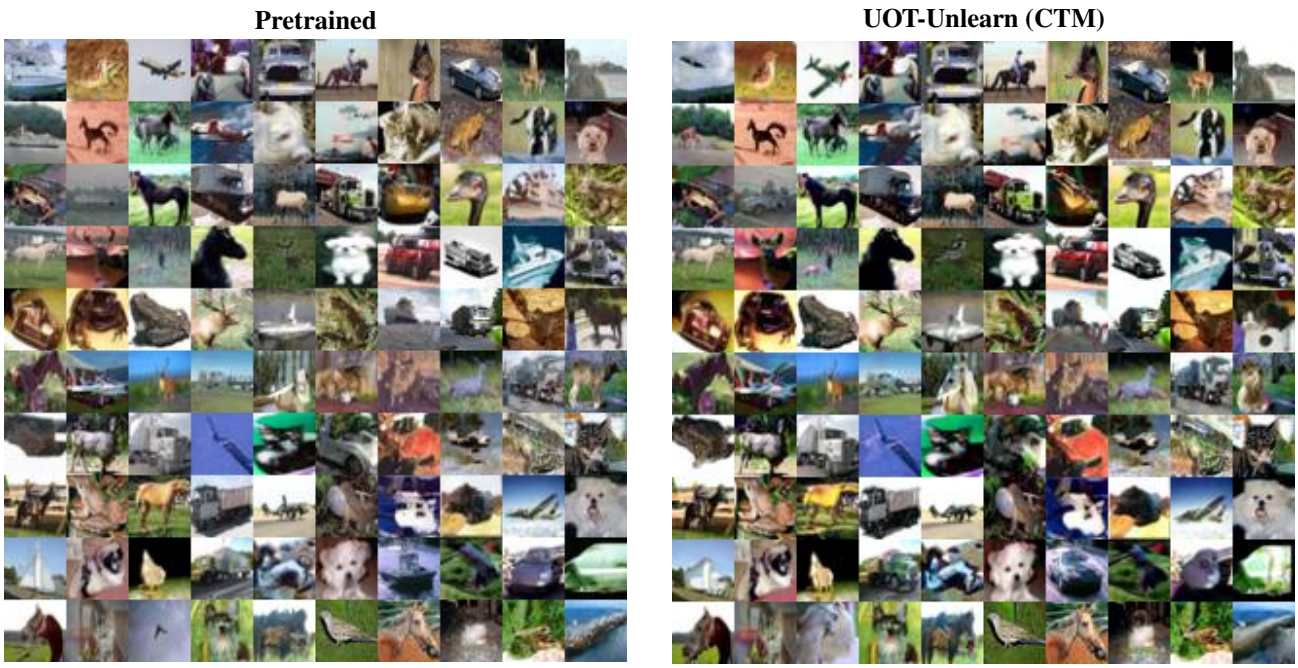


Figure 7. Visual comparison of images generated from the pretrained CTM versus our UOT-Unlearn-applied CTM trained to forget the Ship class (Class 8). The results demonstrate that UOT-Unlearn successfully erases the forget class, seamlessly transitioning it into features of other classes, while strictly preserving the visual quality and structural layout of the remaining retain classes.

Figure 8. 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.

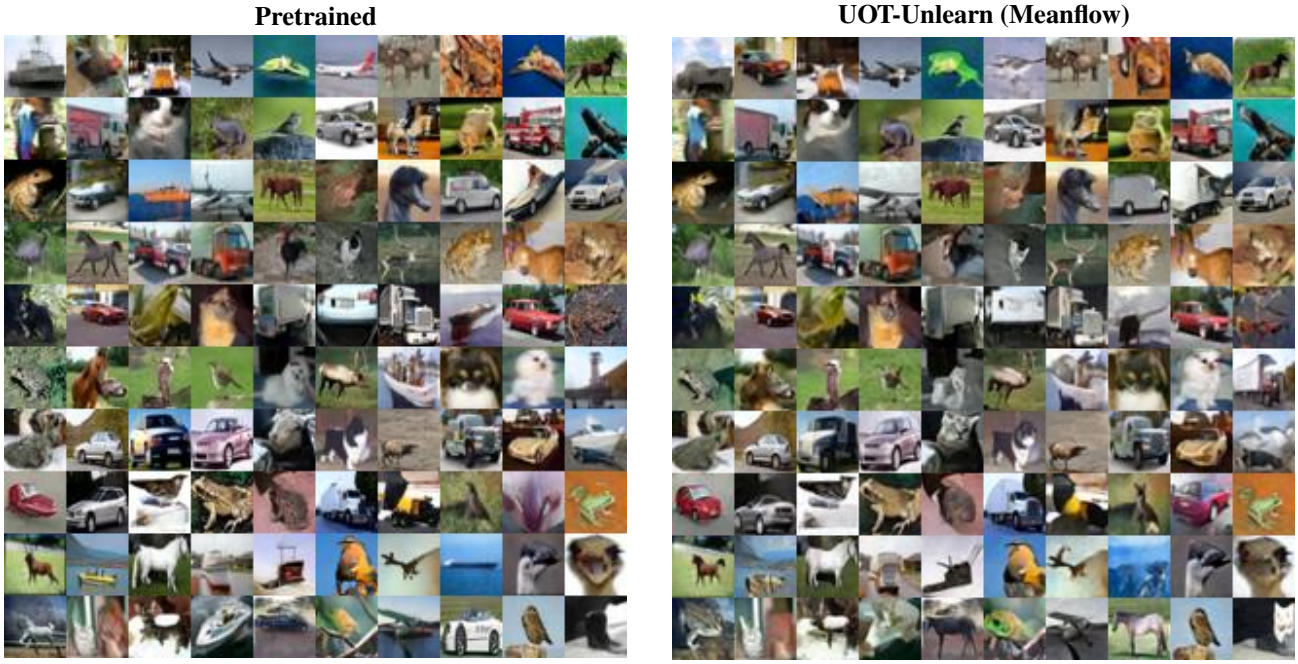
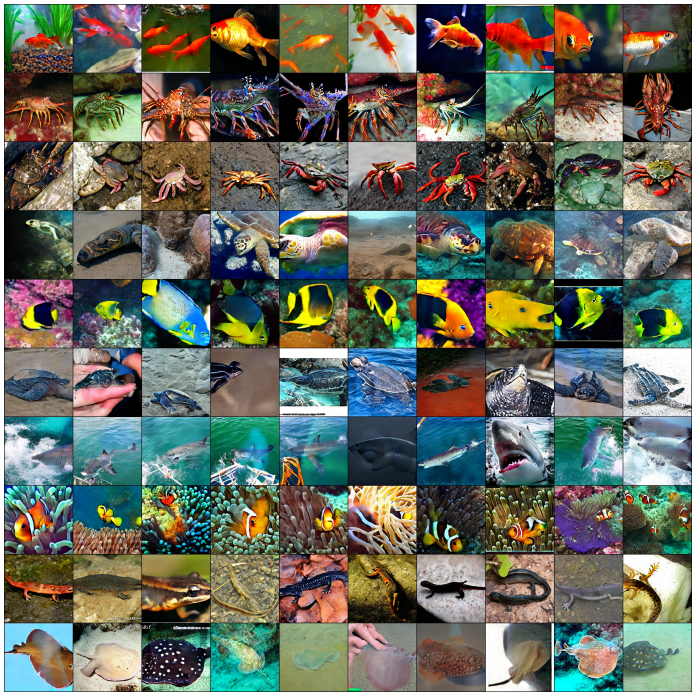
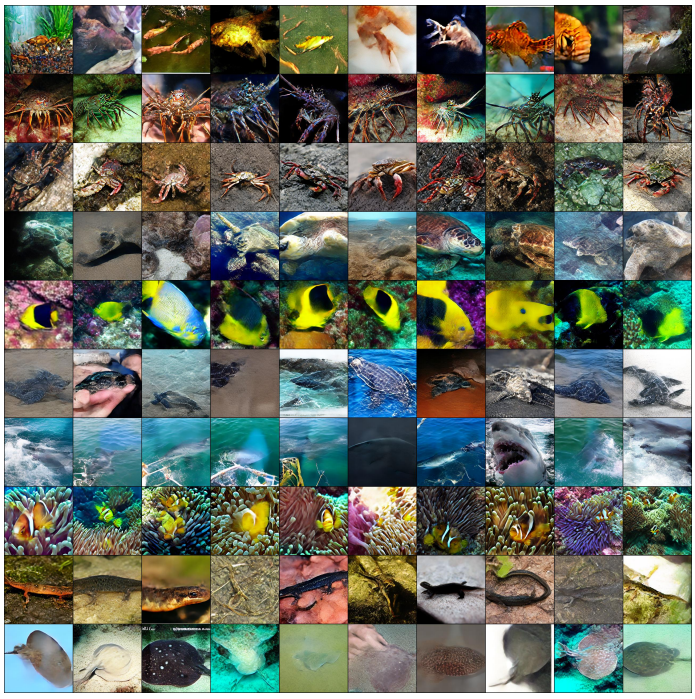


Figure 9. Visual comparison of randomly generated images from the Pretrained model and our UOT-Unlearn (MF) trained to forget the Ship class (Class 8), using the same initial noises.

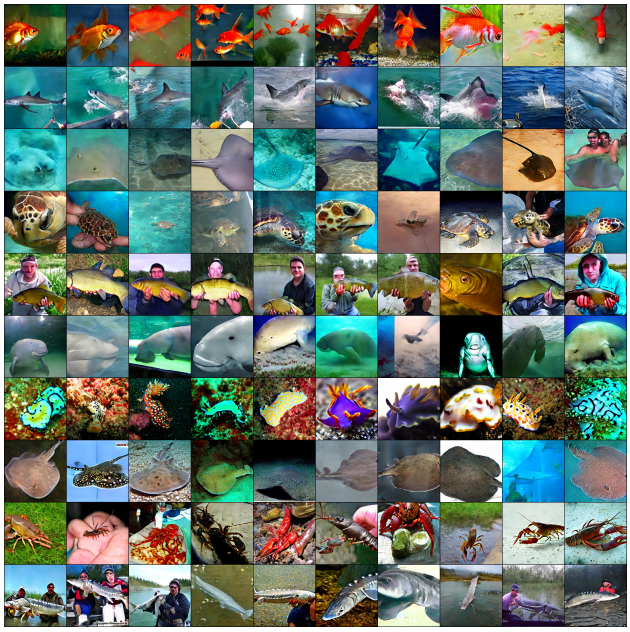


Pretrained

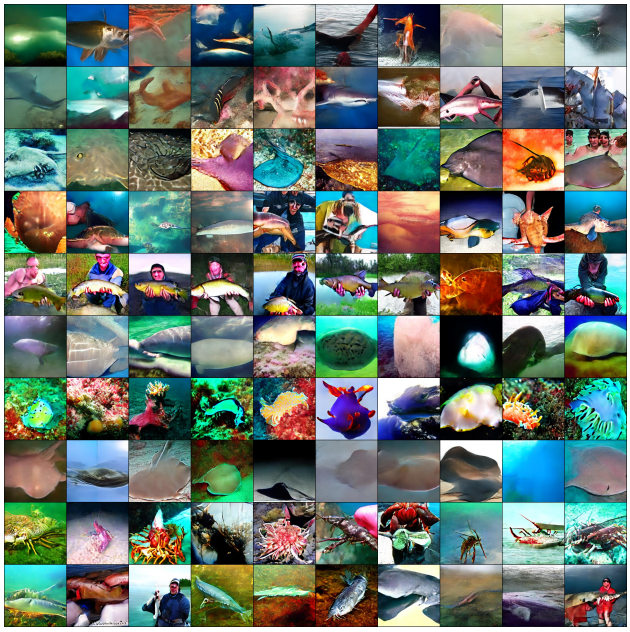


UOT-Unlearn (Meanflow)

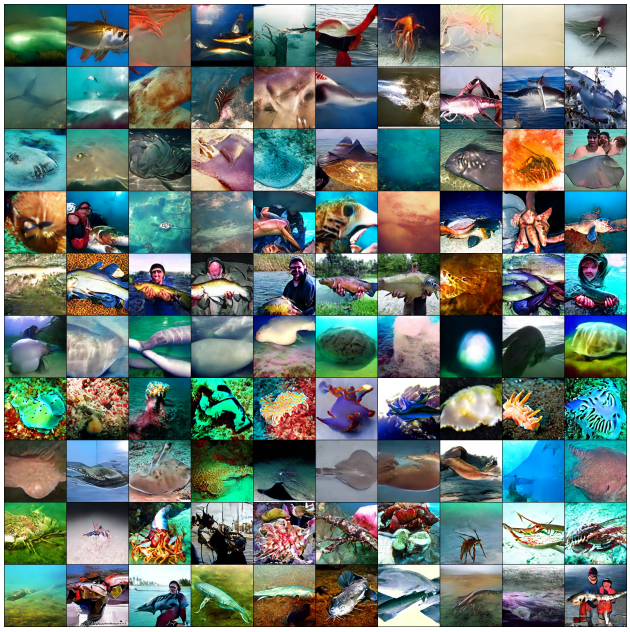
Figure 10. Unlearning results on the ImageNet 256×256 dataset using the MeanFlow (MF) model. The top row of each generated batch displays the Goldfish class (forget class), while the subsequent bottom rows show 9 other aquatic animal classes (retain classes) using the exact same initial noises.



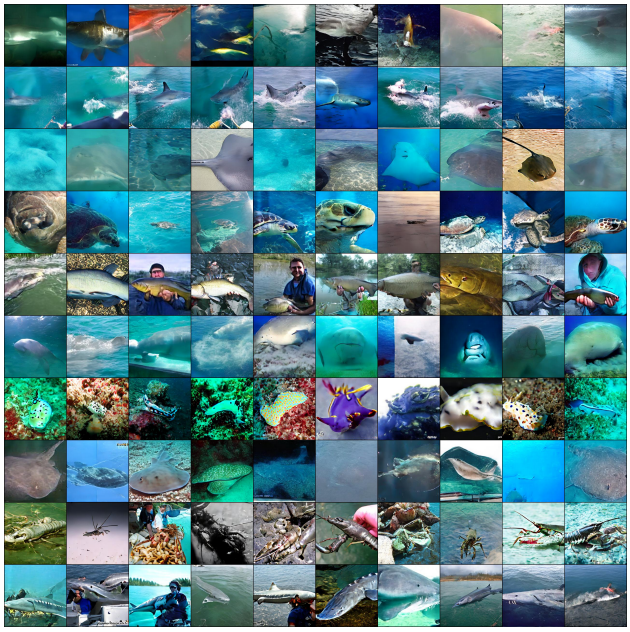
Pretrained



SA



SalUn



VDU

Figure 11. Qualitative comparison of baseline unlearning methods on ImageNet 256×256 using the Meanflow model. The top row of each grid represents the forget class (goldfish), the second row represents the target mapping class (shark), and the subsequent rows show various aquatic retain classes.