DATA UNLEARNING VIA INVERSE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-step matching models, including flow and diffusion models, produce high-quality outputs but incur substantial inference costs and may reproduce unwanted components of their training datasets. We introduce Inverse Distillation Unlearning (IDU), a unified framework that simultaneously distills a teacher multi-step matching model into an efficient one-step student generator and suppresses outputs corresponding to a designated training subset. We first formulate distillation as a min-max objective over a data distribution and then represent this distribution as a mixture of the forget-set and the generated distributions. This allows us to compare this mixture with the teacher's training distribution and recover only the retained data at the optimum. Our method requires only a pretrained full-data teacher and data from the forget set, without access to retained training examples, extra feature extractors or classifiers. Extensive experiments on MNIST and CIFAR-10 datasets under flow-matching and score-based diffusion settings demonstrate that IDU substantially reduces the generation frequency of forgotten classes while preserving generation quality on the retained classes. To the best of our knowledge, IDU is the first unified framework for simultaneous unlearning and distillation in unconditional flow-matching and score-based models.

1 INTRODUCTION

Multi-step diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021), flow (Lipman et al., 2023; Liu et al., 2023), and other matching models (Holderrieth et al., 2024; Gao et al., 2025b), followed by one-step generators (Kim et al., 2024; Zhou et al., 2024; Yin et al., 2024a; Frans et al., 2025), now produce high-quality and diverse samples. Yet their reliance on large, often imperfectly moderated web datasets (Schuhmann et al., 2022) raises privacy, copyright, legal, and ethical concerns (Voigt & Von dem Bussche, 2017; Goldman, 2020): models may memorize and reproduce unwanted training content (Carlini et al., 2023). To avoid such incidents, a critical research direction known as *machine unlearning (MU)* has emerged within the field of trustworthy machine learning (Bourtoule et al., 2021; Nguyen et al., 2025). At its core, MU aims to remove the influence of specific training data, classes, or semantic concepts from trained generative models, while preserving the same generation quality for the remaining data. The MU literature splits into two main paradigms: *data unlearning and class unlearning*.

Data unlearning (Alberti et al., 2025) addresses scenarios where the model is fine-tuned to completely erase the influence of *particular data* for which no class or prompt anchor is available — such as individual faces or Not-Safe-For-Work (NSFW) images. In this setup, the forget data is typically provided as selected samples. Ideally, the goal is to obtain a model as if it were trained only on the remaining data, yet without retraining from scratch. Data unlearning naturally appears in unconditional models, but can also be applied to conditional variants when some data should be erased within a given class.

The key challenge in this setup is that the forget and remaining data are deeply entangled within the model parameters — the forget data cannot be directly prompted during generation. Furthermore, the quality of the remaining data must be preserved, even though its samples may be unavailable. As a result, typical data unlearning algorithms distinguish between remaining and forget data by employing different losses on them (Alberti et al., 2025; Wu et al., 2025; Jiang et al., 2025), constrained optimization (Khalafi et al., 2026), variational framework (Panda et al., 2024), importance sampling (Shi et al., 2026), energy functions (Simone et al., 2025), or transport costs (Choi et al., 2026).

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.

Data unlearning algorithms are well-explored for matching models, with both general frameworks and model-specific solutions (e.g., flow matching (Simone et al., 2025) or score-based models (Jiang et al., 2025)). *Nevertheless, little progress (Choi et al., 2026) has been made toward effective, easy-to-tune, and model-agnostic approaches for one-step generators.*

Class unlearning (Gandikota et al., 2023) aims to remove *all knowledge* pertaining to unwanted classes in the conditional models. More specifically, unlearned models are fine-tuned to output noise or irrelevant data when conditioned on a particular class, such as an unsafe object category, while preserving the generative quality for all other classes. In contrast to data unlearning, the forget data in this setup is typically provided as class labels, without any data samples. Concept unlearning extends this idea to text-to-image models, where the goal is to remove entire concepts or styles (e.g., artistic style, nudity or celebrity's likeness) that may be triggered by textual prompts.

Most class unlearning approaches optimize a combination of two losses: a forget loss that swaps the undesirable class with another one, and a remaining loss that preserves quality for other classes. Various techniques have been developed for designing and combining these losses for diffusion and flow models, including steer-away guidance (Gandikota et al., 2023), Bayesian continual learning (Heng & Soh, 2023), saliency-based weight updates (Fan et al., 2024), cross-attention editing (Gandikota et al., 2024; Lu et al., 2024), attention re-steering (Zhang et al., 2024), pruning (Chavhan et al., 2024), adversarial strategies (Bui et al., 2025), multi-objective optimization to avoid conflicts between loss gradients (Wu et al., 2025), class swapping during distillation (Chen et al., 2025), attention regularization (Gao et al., 2025a; Fan et al., 2026), and unlearning irreversibility techniques (Sharma et al., 2024; Liu & Zhang, 2025; Lu et al., 2025). *However, only a few papers (Chen et al., 2025) tackle class unlearning in one-step models, motivating further research.*

1.1 Contributions

We fill the gap in data unlearning for one-step models and propose our novel **Inverse Distillation Unlearning (IDU)** approach. Our IDU can erase undesired data samples from a pretrained one-step generator or build a new one without them, using only the teacher matching model trained on the original dataset. Our method follows an inverse distillation pipeline and compares the current mixture of the generated and forget data with the teacher's correct one, penalizing the generator for reproducing unwanted samples. Moreover, IDU can work with different matching teachers, such as diffusion or flow models, and requires neither extra training-time feature extractors nor classifiers; its only method-specific trade-off parameter is $\rho \in [0, 1)$.

2 Related Work

2.1 Diffusion, flow and matching models

Denoising diffusion probabilistic model (Ho et al., 2020, **DDPM**) defines a multi-step forward process, mapping data to Gaussian noise, and learns to reverse it via a trained denoiser. Score-based generative model (Song et al., 2021, **SGM**) extends this idea to continuous time and approximates a score function to simulate the reverse SDE. Flow matching (Lipman et al., 2023, **FM**) instead learns an ODE drift that interpolates between data and noise, enabling fewer sampling steps via advanced ODE solvers and flexible interpolations. In all of the above cases, a model must approximate an intractable reverse-process function (denoiser, score, drift, etc.). It is done via available unbiased estimates of this function, conditioned on initial data samples. Since a model is usually matched with these conditional function estimates, we refer to such models collectively as *matching models*.

Formally, a matching model constructs a probability path p_t on the time interval $[0, T]$, transforming the selected data p_0 to noise p_T. This path $p_t(x_t) = \int_{\mathbb{R}^D} p_t(x_t|x_0)p_0(x_0)dx_0$ is built as a mixture of simple conditional paths $p_t(\cdot|x_0)$ conditioned on samples $x_0 \sim p_0$. Then, the standard universal matching (**UM**) loss $\mathcal{L}_{\text{UM}}(f, p_0)$ matches a model $f : [0, T] \times \mathbb{R}^D \to \mathbb{R}^D$ with conditional estimates $f_t^{p_0}(\cdot|x_0)$ at each time t and point $x_t \sim p_t$:

$$\mathcal{L}_{\text{UM}}(f, p_0) := \mathbb{E}_{t, x_0 \sim p_0, x_t \sim p_t(\cdot|x_0)}[\|f_t(x_t) - f_t^{p_0}(x_t|x_0)\|^2]. \tag{1}$$

Here, the notation $\mathbb{E}_t$ hides the time sampling and loss weighting inherent to the given matching model. For example, denoising models recover unnoised samples $f_t^{p_0}(x_t|x_0) = x_0$, score-based

models match the conditional score $f_t^{p_0}(x_t|x_0) = \nabla_{x_t} \ln p_t(x_t|x_0)$, and flow models with the linear interpolation $x_t = (1-t/T)x_0 + (t/T)x_T$ match the conditional drift $f_t^{p_0}(x_t|x_0) = (x_T - x_0)/T = (x_t - x_0)/t$ for $t > 0$.

2.2 Distillation and one-step models

Multi-step sampling makes matching models slower than one-step generators such as VAEs (Kingma & Welling, 2013) and GANs (Goodfellow et al., 2014). Beyond sampling acceleration (Lu et al., 2022; Karras et al., 2024), matching distillation methods (Yin et al., 2024b) train a *one-step generator* G_θ under a *multi-step teacher* f^*: a fake model fits the generated distribution, while the generator reduces its discrepancy from the teacher. Despite differences in model type and discrepancy measure (Yin et al., 2024a; Zhou et al., 2024; Gushchin et al., 2025), these methods admit a common inverse-optimization view (Kornilov et al., 2026). Given $f^* = \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^*)$ trained via UM loss minimization, they recover its data distribution p_0^*. To do this, they optimize the following min-max inverse distillation scheme over trainable generated distribution p_0^θ:

$$\min_\theta \max_f \left\{\mathcal{L}_{\text{UM}}(f^*, p_0^\theta) - \mathcal{L}_{\text{UM}}(f, p_0^\theta)\right\} = \min_\theta \left\{\mathcal{L}_{\text{UM}}(f^*, p_0^\theta) - \min_f \{\mathcal{L}_{\text{UM}}(f, p_0^\theta)\}\right\}. \quad (2)$$

The non-negative difference between losses in this scheme measures how well the teacher fits the current data compared to the best possible fake model. When the teacher data is retrieved, i.e., $p_0^\theta = p_0^*$, the difference becomes 0 and the scheme attains optimum.

2.3 Data unlearning

In data unlearning, we have an original data distribution p_0^*, a forget-data distribution p_0^F whose influence we would like to remove, and the remaining data p_0^R. The ideal solution would be a model trained on the remaining data p^R via the vanilla loss $\mathcal{L}_{\text{vanilla}}$. However, instead of training from scratch, the early works (Golatkar et al., 2020; Thudi et al., 2022; Tang & Khanna, 2026) propose to fine-tune a weighted sum of the forget and remaining losses with a trade-off factor $\rho \in [0, 1)$:

$$\mathcal{L}_{\text{base}} = \rho \cdot \mathcal{L}_{\text{forget}} + (1 - \rho) \cdot \mathcal{L}_{\text{remain}}, \quad (3)$$

where the forget loss $\mathcal{L}_{\text{forget}}$ keeps the model away from the forget data, whereas the remaining loss $\mathcal{L}_{\text{remain}}$ ensures that the quality on the remaining samples does not degrade.

For *unlearning in matching models*, the typical losses are $\mathcal{L}_{\text{vanilla}}(f) = \mathcal{L}_{\text{UM}}(f, p_0^R)$, $\mathcal{L}_{\text{forget}}(f) = -\mathcal{L}_{\text{UM}}(f, p_0^F)$, and $\mathcal{L}_{\text{remain}}(f) = \mathcal{L}_{\text{UM}}(f, p_0^R)$. Many data unlearning methods modify the forget and remaining losses along with the optimization procedure between them. **NegGrad** minimizes only forget loss $\mathcal{L}_{\text{forget}}(f)$. However, this approach often leads to instability and catastrophic forgetting, degrading the overall generation quality. **SA** (Heng & Soh, 2023) adds a computationally heavy Elastic Weight Consolidation penalty to the base loss - the quadratic form of divergence from the initial weights, computed using the Fisher Information Matrix on the current generated data. **SalUn** (Fan et al., 2024) optimizes the base loss and applies a binary gradient mask, selecting only the most significant parameters for forgetting. This mask is calculated by thresholding the gradient magnitudes of the forget loss. **SISS** (Alberti et al., 2025) employs importance sampling to call the model only once per base loss $\mathcal{L}_{\text{base}}(f)$ calculation, but basically does not change the loss structure. **MGSM** (Jiang et al., 2025) incorporates more natural score-function orthogonality instead of ℓ_2-loss for the forget part. **EraseDiff** (Wu et al., 2025) utilizes random-noise matching loss on forget samples and employs constrained optimization to smoothly merge the gradient directions of the losses. **Retrack** (Shi et al., 2026) equips the remaining loss $\mathcal{L}_{\text{remain}}(f)$ with the nearest-neighbor importance weighting from the forget subset. The work (Khalafi et al., 2026) optimizes KL divergence between noising processes, also under the constrained optimization formulation.

Other methods use various types of guidance to avoid the forget data. **VDU** (Panda et al., 2024) uses a variational inference framework with a plasticity inducer for reducing the likelihood of unwanted data and a stability regularizer for quality preservation. For flow models, **ContinualFlow** (Simone et al., 2025) applies energy-function weighting to the original loss to suppress unwanted data.

One-step model unlearning. **UOT-Unlearn** (Choi et al., 2026) uses unbalanced optimal transport to shift a pretrained one-step generator away from unwanted samples. Its cost function requires a precomputed feature extractor and hyperparameter tuning rather than a trained teacher. However, the OT framework has limited scalability and generation diversity, compared with matching models

distillation. The same work adapts VDU, SalUn, and SA to consistency models (Kim et al., 2024; Geng et al., 2025; Frans et al., 2025); these adaptations use a classifier to identify forget samples and do not directly extend to non-self-consistent generators.

2.4 Class unlearning

Class unlearning aims to erase entire unwanted classes from a conditional model while keeping other classes intact. Unlike specific data samples, whose influence in an unconditional model is difficult to trace, classes in a conditional model can be efficiently prompted and distinguished from one another. For the same reason, the majority of class-forgetting methods are data-free. These features enable a variety of methods for *matching model unlearning* that bind the attributes of the unwanted classes to completely different entities, especially within the attention or inference structure (Gandikota et al., 2023; 2024; Lu et al., 2024; Zhang et al., 2024; Gao et al., 2025a; Fan et al., 2026). Such methods are not always adaptable to data unlearning; nevertheless, they often use the same combination of the forget and remaining losses: the forget loss changes the model's behavior on unwanted classes, while the remaining loss preserves it on the others. For example, SA (Heng & Soh, 2023), SalUn (Fan et al., 2024), and EraseDiff (Wu et al., 2025) can be leveraged for both data and class unlearning tasks.

One-step model unlearning. **SFD** (Chen et al., 2025) performs class forgetting during conditional inverse distillation (2) by replacing the teacher score for forgotten classes with a safe-class score in the generator loss; its full generator losses are given in Appendix A.3.

3 Inverse Distillation Unlearning

3.1 Method Description

Setup and preliminaries. In data unlearning, we are given a forget-data distribution p_0^F that we would like to remove from the original distribution p_0^*, so that only the remaining (or retained) data p_0^R is preserved:

$$p_0^* = \pi\, p_0^F + (1-\pi)\, p_0^R, \tag{4}$$

where $\pi \in [0, 1)$ is the proportion of the forget data. In our setup, neither the original nor the remaining data is available, but we do have access to a teacher model $f^* = \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^*)$ of an arbitrary matching type, trained on the original data. We aim to train a one-step generator $G_\theta : \mathcal{Z} \to \mathbb{R}^D$ with parameters θ that will eventually reproduce only the remaining data p_0^R. The generator maps the latent distribution $p_{\mathcal{Z}}$ to a distribution p_0^θ and can be pretrained or initialized with a one-step teacher inference scheme.

The standard way to distill the data p_0^*, stored inside the teacher model f^*, into the trainable distribution p_0 is to apply the inverse distillation scheme (2). This min-max scheme reverses the forward minimization problem for obtaining the teacher from the fixed input data, i.e., it retrieves the data from which the fixed teacher was obtained:

$$\min_{p_0} \max_f \{\mathcal{L}_{\text{UM}}(f^*, p_0) - \mathcal{L}_{\text{UM}}(f, p_0)\} \sim \min_{p_0} \{\underbrace{\mathcal{L}_{\text{UM}}(f^*, p_0) - \min_f \{\mathcal{L}_{\text{UM}}(f, p_0)\}}_{\geq 0}\}. \tag{5}$$

The non-negative difference between losses in this scheme measures how well the teacher fits the current data compared to the best possible fake model. For the teacher data $p_0 = p_0^*$, the optimum is attained with the zero difference.

Our approach. We build our **Inverse Distillation Unlearning (IDU)** method as follows: we run the inverse distillation scheme (5) but parametrize the optimized distribution p_0 as the mixed data $p_0 = \rho\, p_0^F + (1-\rho)\, p_0^\theta =: p_0^{\text{mix}}$, similar to the data mix (4) with the proportion $\rho \in [0, 1)$, where we substitute the desired remaining data p_0^R with the generated one p_0^θ. Thus, the optimal generator with the right proportion $\rho = \pi$ has to learn only the remaining data in order to recover the full teacher distribution, since the forget component is already accounted for by the forget samples. The following theorem formalizes the generator's forgetting property. The proof is given in Appendix A.1.

Theorem 1 (IDU's forgetting property) *Optimization of IDU loss (6) with $\rho = \pi$ retrieves only the remaining data, i.e., the optimal generator parameters θ_{opt} yield $p_0^{\theta_{opt}} = p_0^R$.*

More specifically, we optimize the following *min-max IDU objective* $\mathcal{L}_{\text{IDU}}(f, p_0^\theta)$ over generator parameters θ and fake model f:

$$\begin{aligned}\mathcal{L}_{\text{IDU}}(f, p_0^\theta) &:= \mathcal{L}_{\text{UM}}(f^*, p_0^{\text{mix}}) - \mathcal{L}_{\text{UM}}(f, p_0^{\text{mix}}) \\ &= \mathcal{L}_{\text{UM}}(f^*, \rho\, p_0^F + (1-\rho)\, p_0^\theta) - \mathcal{L}_{\text{UM}}(f, \rho\, p_0^F + (1-\rho)\, p_0^\theta) \\ &= \rho \cdot [\mathcal{L}_{\text{UM}}(f^*, p_0^F) - \mathcal{L}_{\text{UM}}(f, p_0^F)] + (1-\rho) \cdot [\mathcal{L}_{\text{UM}}(f^*, p_0^\theta) - \mathcal{L}_{\text{UM}}(f, p_0^\theta)], \quad (6)\end{aligned}$$

where the second equality holds because the UM losses (1) are linear in the input data, which appears only inside a mathematical expectation.

3.2 Practical Details

Optimization procedure. We alternate between two steps to optimize our min-max IDU loss (6):

1) First, we update the fake model f via the fake loss $\mathcal{L}_{\text{IDU-fake}}(f)$:

$$\begin{aligned}&\mathcal{L}_{\text{IDU-fake}}(f) = \rho \cdot \mathcal{L}_{\text{UM}}(f, p_0^F) + (1-\rho) \cdot \mathcal{L}_{\text{UM}}(f, p_0^\theta) \\ &= \mathbb{E}_{\substack{t, x_0^\theta \sim p_0^\theta, x_t^\theta \sim p_t^\theta(\cdot|x_0^\theta) \\ x_0^F \sim p_0^F, x_t^F \sim p_t^F(\cdot|x_0^F)}} [\rho \|f_t(x_t^F) - f_t^F(x_t^F|x_0^F)\|^2 + (1-\rho)\|f_t(x_t^\theta) - f_t^\theta(x_t^\theta|x_0^\theta)\|^2], \quad (7)\end{aligned}$$

where $p_t^\theta(\cdot|x_0^\theta)$ and $p_t^F(\cdot|x_0^F)$ are the conditional forward noising processes built on the generated and forget data with the corresponding conditional estimates $f_t^\theta(x_t^\theta|x_0^\theta)$ and $f_t^F(x_t^F|x_0^F)$.

2) Next, we update the generator parameters θ with a fixed fake model f via the generator loss $\mathcal{L}_{\text{IDU-gen}}(p_0^\theta)$; however, instead of the default loss $\mathcal{L}_{\text{IDU-gen}}(p_0^\theta) = \mathcal{L}_{\text{UM}}(f^*, p_0^\theta) - \mathcal{L}_{\text{UM}}(f, p_0^\theta)$, we use its modified version, proposed in the Score identity Distillation (SiD) framework (Zhou et al., 2024):

$$\mathcal{L}_{\text{IDU-gen}}(p_0^\theta) = 2\mathbb{E}_{\substack{t, x_0^\theta \sim p_0^\theta, \\ x_t^\theta \sim p_t^\theta(\cdot|x_0^\theta)}} [\langle f_t^*(x_t^\theta) - f_t(x_t^\theta), f_t^*(x_t^\theta) - f_t^\theta(x_t^\theta|x_0^\theta)\rangle - \alpha_{\text{SiD}} \|f_t^*(x_t^\theta) - f_t(x_t^\theta)\|^2],$$

where we heuristically scale the second term by exactly α_{SiD} times. This scale factor α_{SiD} is usually taken from the range $\alpha_{\text{SiD}} \in [0.5, 1.2]$. For example, in case $\alpha_{\text{SiD}} = 0.5$, we end up with the default theoretical loss, whereas greater factor values can yield better performance in practice. Nevertheless, the optimal value depends strongly on the matching model type and neural network architecture.

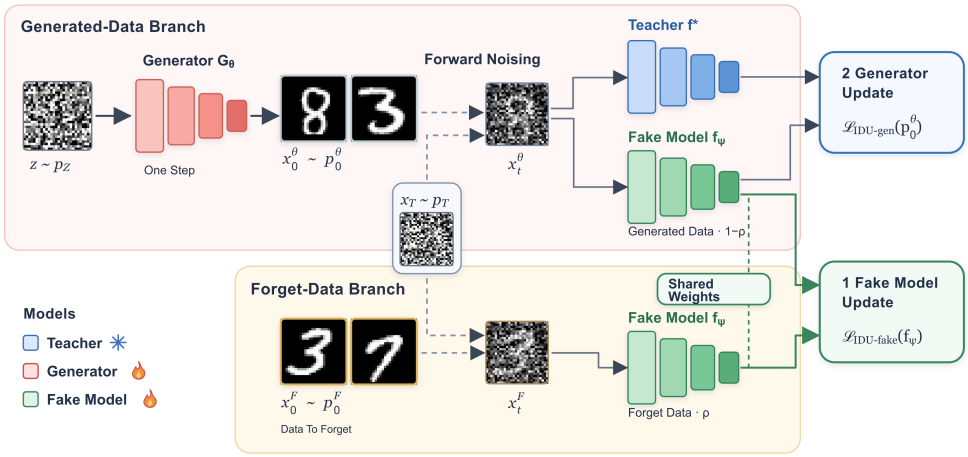


Figure 1: **Pipeline of our IDU framework**. Forget and generated samples form two forward-noising branches. The framework alternates between two steps: first, update the fake model on both branches with weights ρ and $1-\rho$ to detect generations similar to the forget samples, then update the generator to suppress these generations using the frozen teacher and the fake model.

Algorithm pseudocode. In Algorithm 1, we provide IDU pseudocode for the flow-matching setting. The general IDU training pipeline is illustrated in Figure 1.

Algorithm 1 Inverse Distillation Unlearning

Input: teacher model f^*, generator G_θ (pretrained or initialized by the teacher), fake model f_ψ, forget data p_0^F, forgetting scale $\rho \in [0, 1)$, SiD scale $\alpha_{\text{SiD}} \in [0.5, 1.2]$, number of iterations N, batch size B, latent distribution $p_{\mathcal{Z}}$, noise distribution p_T.

1: **for** $n = 0, \dots, N-1$ **do**

2: Sample generated and noise batches $\{x^\theta_{0,i} = G_\theta(z_i)\}_{i=1}^B$, $z_i \sim p_{\mathcal{Z}}$ and $\{x_{T,i}\}_{i=1}^B \sim p_T$;

3: Sample times $\{t_i\}_{i=1}^B$ and noised samples $\{x^\theta_{t_i,i}\}_{i=1}^B$ according to the model type;
For flow models: $x^\theta_{t_i,i} = (1 - t_i/T) \cdot x^\theta_{0,i} + t_i/T \cdot x_{T,i}$;

4: Sample forget data batch $\{x^F_{0,i}\}_{i=1}^B \sim p_0^F$ and noised forget samples $\{x^F_{t_i,i}\}_{i=1}^B$;
For flow models: $x^F_{t_i,i} = (1 - t_i/T) \cdot x^F_{0,i} + t_i/T \cdot x_{T,i}$;

5: Compute both matching targets $s^\theta_{t_i,i} = f^\theta_{t_i}(x^\theta_{t_i,i}|x^\theta_{0,i})$ and $s^F_{t_i,i} = f^F_{t_i}(x^F_{t_i,i}|x^F_{0,i})$;
For flow models: $s^\theta_{t_i,i} = (x_{T,i} - x^\theta_{0,i})/T$ and $s^F_{t_i,i} = (x_{T,i} - x^F_{0,i})/T$;

6: Update fake model parameters ψ via loss:

$$\frac{1}{B}\sum_{i=1}^{B}\left[\rho\|f_{\psi,t_i}(x^F_{t_i,i}) - s^F_{t_i,i}\|^2 + (1-\rho)\|f_{\psi,t_i}(x^\theta_{t_i,i}) - s^\theta_{t_i,i}\|^2\right];$$

7: Update generator parameters θ via loss:

$$\frac{1}{B}\sum_{i=1}^{B}[2\langle f^*_{t_i}(x^\theta_{t_i,i}) - f_{\psi,t_i}(x^\theta_{t_i,i}), f^*_{t_i}(x^\theta_{t_i,i}) - s^\theta_{t_i,i}\rangle - 2\cdot\alpha_{\text{SiD}}\|f^*_{t_i}(x^\theta_{t_i,i}) - f_{\psi,t_i}(x^\theta_{t_i,i})\|^2];$$

8: **end for**

Hyperparameters. The only new hyperparameter introduced by our IDU method is the scale factor $\rho \in [0, 1)$ which controls the strength of forgetting applied to the selected data. The theoretically justified value of $\rho = \pi$ can be approximately derived from the data split (4) as the proportion of the forget data within the overall dataset. Nevertheless, we still recommend trying other values of ρ in practice to find the best trade-off between forgetting rate and retention quality. We demonstrate this trade-off for FM and SiD in Table 3. For the backbone-specific distillation scale, we use $\alpha_{\text{SiD}} = 0.5$ for FM and $\alpha_{\text{SiD}} = 1.2$ for SiD, following the corresponding original distillation recipes (Zhou et al., 2024; Kornilov et al., 2026). Further optimization details are provided in Appendix B.5.

4 Experiments

Experimental setup. We evaluate IDU with flow matching (FM) (Tong et al., 2023) and the EDM-VP score-based backbone (Karras et al., 2024) used by Score identity Distillation (SiD) (Zhou et al., 2024), each on MNIST and CIFAR-10. The main experiments forget digits $3, 7$ or CIFAR-10 classes $1, 9$ (automobile and truck); the latter pair tests jointly removing related modes. Class membership makes forgetting measurable, but IDU receives only forget samples, not labels or prompts. From each frozen full-data teacher, we run ordinary distillation and IDU, with the former verifying that the same framework also recovers a standard one-step generator when no forget data are mixed in. IDU optimization accesses only the teacher and forget samples. Retained images and evaluation classifiers are never supplied to the training objective. Architectures, samplers, and hyperparameters are detailed in Appendices B.1, B.2, and B.5.

Evaluation protocol and metrics. We compute FID against the full training set for full-data teachers and ordinary distillation, and against retained data for IDU and retraining (*Retain FID*). A teacher trained on retained data and its distilled student serve as retraining baselines. We also report the *forgotten-class generation rate (FGR)*: the percentage of 50,000 generated images assigned to a forgotten class by an off-the-shelf classifier. Classifier and FID protocols are in Appendices B.3 and B.4. Unless noted, values are means and standard deviations over five evaluations; marked CIFAR-10 score-based values follow (Zhou et al., 2024).

Table 1: Main FM/SiD results on MNIST and CIFAR-10. FID uses full-data references for Pretrain/Pure distillation and retained-data references for Retrain/IDU; FGR is reported per forgotten class. Values are mean ± standard deviation over five runs, except † values from (Zhou et al., 2024).

	MNIST		CIFAR-10	
Mode	FID ↓	FGR (%) ↓ Class 3 / Class 7	FID ↓	FGR (%) ↓ Class 1 / Class 9
FM				
Pretrain	0.88 ± 0.01	10.40 ± 0.12 10.10 ± 0.26	3.66 ± 0.03	12.50 ± 0.08 11.18 ± 0.12
Pure distillation	3.23 ± 0.03	10.50 ± 0.15 10.53 ± 0.09	4.35 ± 0.05	7.58 ± 0.09 8.75 ± 0.21
Forgotten classes	$\{3, 7\}$		$\{1, 9\}$	
Retrain teacher	0.340 ± 0.001	—	3.93 ± 0.02	—
Retrain distillation	5.72 ± 0.12	—	5.12 ± 0.05	—
IDU ($\rho_{\text{MNIST}} = 0.4$) ($\rho_{\text{CIFAR-10}} = 0.6$)	3.57 ± 0.03	0.16 ± 0.01 0.16 ± 0.01	5.81 ± 0.05	0.56 ± 0.05 0.39 ± 0.03
SiD				
Pretrain	1.35 ± 0.02	10.99 ± 0.14 9.65 ± 0.22	$1.97^{\dagger}$	11.13 ± 0.12 10.04 ± 0.15
Pure distillation	1.12 ± 0.01	10.51 ± 0.12 10.07 ± 0.24	$1.92 \pm 0.02^{\dagger}$	10.11 ± 0.14 10.66 ± 0.14
Forgotten classes	$\{3, 7\}$		$\{1, 9\}$	
Retrain teacher	0.85 ± 0.02	—	2.26 ± 0.02	—
Retrain distillation	1.20 ± 0.02	—	2.82 ± 0.02	—
IDU ($\rho = 0.2$)	1.65 ± 0.02	0.63 ± 0.03 0.32 ± 0.02	3.15 ± 0.03	1.11 ± 0.06 1.25 ± 0.06

Main results. Table 1 shows that IDU suppresses both forgotten modes under all four dataset–backbone combinations. In the FM experiments, every forgotten-class FGR is at most 0.56%; in the SiD experiments, it is at most 1.25%. IDU also retains generation quality close to the corresponding retrain-distillation oracle in three settings and improves on that reference for FM on MNIST, indicating no substantial Retain FID degradation relative to target-matched retraining. The FM/MNIST oracle comparison warrants caution owing to sensitivity of the retained-only retraining baseline; see Appendix B.9. Visual results with Teacher–IDU sample grids appear in Appendix C. Fine-tuning experiments on the purely distilled generators achieve similar metrics; see Appendix B.7.

Robustness across forget classes. We also evaluate single-class forgetting for all MNIST classes with FM and all CIFAR-10 classes with FM and SiD (Table 2). Across these ten tasks per setting, mean Retain FID/FGR is $4.54/0.16\%$ for FM–MNIST, $7.73/0.84\%$ for FM–CIFAR-10, and $3.61/0.75\%$ for SiD–CIFAR-10. The per-class results show that low FGR is not confined to the pairs used in the main experiments.

Effect of the forgetting weight. Table 3 varies the forget-mixture weight ρ on CIFAR-10 for both FM and SiD. For FM, $\rho = 0.9$ diverges immediately. Among the stable runs, low ρ preserves FID but leaves higher FGR, whereas $\rho = 0.8$ improves forgetting at a substantial FID cost. We therefore use $\rho = 0.6$ as the best empirical balance for FM on CIFAR-10. The FM coefficient for MNIST, $\rho = 0.4$, is selected by the same quality–forgetting criterion. For SiD, increasing ρ from 0.2 to 0.8 progressively worsens Retain FID, whereas the two class-wise FGR values vary non-monotonically. At $\rho = 0.9$, SiD does not diverge, but its FGRs (12.05% and 9.55%) approach those of the full-data teacher, while Retain FID rises to 10.36. Thus, excessive ρ can degrade fidelity without forgetting.

Table 2: Single-class robustness of IDU with FM on MNIST and CIFAR-10 and with SiD on CIFAR-10. The final row averages the per-class means across the ten forget-class experiments.

Forgotten Class	FM MNIST		FM CIFAR-10		SiD CIFAR-10	
	Retain FID ↓	FGR (%) ↓	Retain FID ↓	FGR (%) ↓	Retain FID ↓	FGR (%) ↓
0	5.05 ± 0.03	0.10 ± 0.01	6.52 ± 0.06	0.76 ± 0.03	3.54 ± 0.02	0.58 ± 0.03
1	3.46 ± 0.04	0.03 ± 0.01	6.45 ± 0.09	1.06 ± 0.07	2.96 ± 0.01	0.61 ± 0.03
2	3.84 ± 0.02	0.12 ± 0.01	7.83 ± 0.05	0.73 ± 0.04	4.07 ± 0.05	0.87 ± 0.04
3	5.23 ± 0.03	0.10 ± 0.01	7.82 ± 0.08	1.51 ± 0.04	4.10 ± 0.02	1.11 ± 0.02
4	4.71 ± 0.03	0.13 ± 0.02	8.53 ± 0.08	0.82 ± 0.04	4.13 ± 0.03	0.78 ± 0.05
5	4.37 ± 0.06	0.17 ± 0.02	9.70 ± 0.05	1.18 ± 0.06	4.21 ± 0.03	0.88 ± 0.04
6	4.39 ± 0.04	0.07 ± 0.01	8.45 ± 0.09	0.23 ± 0.02	4.16 ± 0.05	0.53 ± 0.03
7	5.17 ± 0.04	0.15 ± 0.01	5.82 ± 0.06	1.23 ± 0.05	2.87 ± 0.02	0.90 ± 0.05
8	4.22 ± 0.04	0.24 ± 0.03	8.88 ± 0.13	0.39 ± 0.02	3.22 ± 0.02	0.54 ± 0.05
9	4.93 ± 0.05	0.45 ± 0.03	7.26 ± 0.05	0.47 ± 0.03	2.86 ± 0.02	0.72 ± 0.01
Mean	4.54	0.16	7.73	0.84	3.61	0.75

Table 3: Effect of the forget-mixture weight ρ on CIFAR-10 for FM- and SiD-based IDU when jointly forgetting automobile (class 1) and truck (class 9). The selected configuration for each setup is bold; dashes denote settings not evaluated with SiD. FM training at $\rho = 0.9$ diverges immediately.

Flow Matching				**SiD**			
ρ	FID ↓	FGR 1 (%) ↓	FGR 9 (%) ↓	ρ	FID ↓	FGR 1 (%) ↓	FGR 9 (%) ↓
0.05	6.72 ± 0.07	3.04 ± 0.08	5.02 ± 0.08			—	
0.1	6.21 ± 0.05	3.75 ± 0.09	3.80 ± 0.08			—	
0.2	5.64 ± 0.04	3.35 ± 0.03	3.04 ± 0.10	0.2	$\mathbf{3.15 \pm 0.03}$	$\mathbf{1.11 \pm 0.06}$	$\mathbf{1.25 \pm 0.06}$
0.4	5.87 ± 0.07	0.62 ± 0.06	0.48 ± 0.02	0.4	3.53 ± 0.04	0.74 ± 0.05	0.60 ± 0.01
0.6	$\mathbf{5.81 \pm 0.05}$	$\mathbf{0.56 \pm 0.05}$	$\mathbf{0.39 \pm 0.03}$	0.6	4.68 ± 0.06	1.16 ± 0.05	0.76 ± 0.02
0.8	8.59 ± 0.11	0.38 ± 0.03	0.32 ± 0.03	0.8	6.29 ± 0.07	0.81 ± 0.05	0.58 ± 0.02
0.9		Diverged		0.9	10.36 ± 0.10	12.05 ± 0.18	9.55 ± 0.12

5 Discussion and Comparison

How does our method work? Our IDU leverages the teacher as guidance to steer generations away from unwanted data. Specifically, the IDU's generator loss is a tractable form of the squared norm of the difference between the teacher f^* and the fake model f^{mix} trained on the mixed forget and generated data $p_0^{\text{mix}} := \rho \cdot p_0^F + (1-\rho) \cdot p_0^\theta$; see Appendix A.2:

$$\mathcal{L}_{\text{IDU}}(f^{\text{mix}}, p_0^\theta) = \mathbb{E}_{t, x_t \sim p_t^{\text{mix}}}[\left\| f_t^*(x_t) - f_t^{\text{mix}}(x_t) \right\|^2], \quad f^{\text{mix}} := \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^{\text{mix}}).$$

Since the supplied forget samples already cover the forget component of p_0^{mix}, further forget samples generated by G_θ make the fake model depart from the teacher. Minimizing the generator loss counteracts this excess while also discouraging low-quality retained generations. Similar ideas are proposed for distillation acceleration in (Kornilov et al., 2026). There, the authors utilize real data to push the generator toward the teacher faster, whereas we use forget data to steer the model away.

Training and evaluation requirements. IDU training uses the frozen teacher and forget samples only; retained images and external classifiers enter neither its losses nor its gradient updates. On public MNIST and CIFAR-10, retained-data FID and classifier-based FGR provide direct, reproducible measurements. We use these metrics to select ρ; without the retained data, ρ can be chosen from a wide range of values, starting from the forget-set proportion, and still avoid significant degradation. However, the ablations show that larger values do not necessarily improve forgetting and may degrade fidelity. If direct evaluation is critical, retained data can be selected from the teacher samples via classification or manual selection.

Data unlearning methods. Our IDU is suitable for both the unlearning of the multi-step teacher matching models (with additional distillation) and the unlearning of the pretrained generators.

We begin by comparing the *unlearning of matching models*. ContinualFlow (Simone et al., 2025) uses an energy function to guide a teacher flow model away from forget data, but it is limited to flow matching and relies on an image classifier to calculate the energy. Other methods, such as NegGrad, SA (Heng & Soh, 2023), SalUn (Fan et al., 2024), SISS (Alberti et al., 2025), Retrack (Shi et al., 2026), MGSM (Jiang et al., 2025), EraseDiff (Wu et al., 2025), and VDU (Panda et al., 2024), unlearn teacher models using a combination of forget and remaining losses (3). Although our IDU also optimizes a similar combination of losses within the fake model, our working principle is different. The forget and remaining losses in other methods are trained adversarially: the former degrades quality on the forget dataset, while the latter preserves it on the remaining one. This is why such methods often resort to constrained optimization or more stable losses. In our IDU, by contrast, the forget and generated data distributions are modeled jointly as a mixture. As a result, we avoid multi-objective optimization and train our losses coherently. Moreover, distilled generators provide efficient one-step inference rather than the multi-step sampling for the teacher.

For pretrained generators, UOT-Unlearn (Choi et al., 2026) employs a cost function from unbalanced optimal transport to guide the unlearning. This guidance requires an additional feature extractor and manual cost-function tuning instead of a teacher model. Our IDU uses more complex teacher guidance but yields better forgetting and retaining metrics. As the UOT-Unlearn code is unavailable, we compare against their best unlearned CTM model with the SiD architecture and similar initial FID of 1.73. For CIFAR-10 unlearning classes 1, 6, and 8, their Retain FIDs (↓) are (9.90, 5.11, 5.88) versus our (2.96, 4.16, 3.22), and their FGRs (%, ↓) are around (2, 0.9, 1.5) versus our (0.61, 0.53, 0.54). Many of the above matching model unlearning methods can be generalized to one-step consistency models. These models enforce self-consistency along the generation trajectory by mapping any point directly to the start. Thus, they can optimize different losses on the forget and remaining data. For other one-step model types that do not take data samples as input, such as GANs, OT, and distillation, this generalization does not work. Moreover, such methods lag significantly behind in terms of retaining and forgetting (see Table 2 in (Choi et al., 2026)). In contrast, our IDU does not require self-consistency or a particular generator model type.

Class unlearning methods. The most relevant SFD approach (Chen et al., 2025) also performs unlearning during distillation but operates in the class unlearning setup. This method focuses on conditional models, where the data to erase is prompted through the input labels rather than through given samples. In contrast, our IDU can erase any part of training data, offering more flexible forgetting opportunities. The working mechanisms also differ dramatically. SFD modifies the generator loss (10), writing safe teacher information into the generator's unwanted classes. We modify the fake model loss to make it remember the mixture of generated and forget data and then compare it with the teacher's correct one, penalizing the generator for reproducing unwanted samples. In the class-defined experiments reported in Tables 1, 2, and 3, the generator is never prompted with a class label: class annotations are used only to construct the forget subset and to evaluate FGR. Nevertheless, under the same conditions, our IDU achieves comparable results. In the CIFAR-10 class-0 forgetting experiment, SFD attains a Retain FID (↓) of around 3.1 and an FGR (%, ↓) of 0.36 (Figure 5 (Chen et al., 2025)), versus 3.54 and 0.58 for our method (Table 2).

Optimization stability across backbones. In extended FM runs on MNIST and CIFAR-10, IDU often suppresses the target classes early, but their generation frequency can rise again after roughly 20k generator updates; the onset depends on ρ. FM therefore requires joint selection of ρ and checkpoint. We did not observe this reversal over the evaluated SiD training horizon. This appears to be a limitation of the current FM instantiation, not of the IDU objective; larger, more stable FM architectures may allow ρ to approach the forget-set proportion, as it does for SiD, where $\rho = 0.2$ matches two forgotten classes out of ten.

AI USE STATEMENT

Generative AI tools were used solely to improve grammar, clarity, concision, and academic style during manuscript preparation. They were not used to formulate the research problem, develop the method, design or execute experiments, generate or analyze results, or determine the scientific claims and conclusions. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content of this work.

REFERENCES

Silas Alberti, Kenan Hasanaliyev, Manav Shah, and Stefano Ermon. Data unlearning in diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SuHScQv5gP.

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.

Anh Bui, Thuy-Trang Vu, Long Vuong, Trung Le, Paul Montague, Tamas Abraham, Junae Kim, and Dinh Phung. Fantastic targets for concept erasure in diffusion models and where to find them. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 64032–64074, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a10946e1f46e1ffc0daf37cb2abfdcad-Paper-Conference.pdf.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Ruchika Chavhan, Da Li, and Timothy Hospedales. Conceptprune: Concept editing in diffusion models via skilled neuron pruning. *arXiv preprint arXiv:2405.19237*, 2024.

Tianqi Chen, Shujian Zhang, and Mingyuan Zhou. Score forgetting distillation: A swift, data-free method for machine unlearning in diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=gjwhDHeAsz.

Hyundo Choi, Junhyeong An, Jinseong Park, and Jaewoong Choi. Unlearning for one-step generative models via unbalanced optimal transport. In *ICML 2026 Workshop on Foundations of Deep Generative Models (FoGen)*, 2026. URL https://arxiv.org/abs/2603.16489.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, volume 2024, pp. 53643–53673, 2024.

Zhaoxin Fan, Nanxiang Jiang, Daiheng Gao, Shiji Zhou, and Wenjun Wu. Eraseanything++: Enabling concept erasure in rectified flow transformers leveraging multi-object optimization, 2026. URL https://arxiv.org/abs/2603.00978.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5099–5108. IEEE, 2024.

Daiheng Gao, Shilin Lu, Wenbo Zhou, Jiaming Chu, Jie Zhang, Mengxi Jia, Bang Zhang, Zhaoxin Fan, and Weiming Zhang. EraseAnything: Enabling concept erasure in rectified flow transformers. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18470–18494. PMLR, 13–19 Jul 2025a. URL https://proceedings.mlr.press/v267/gao25j.html.

Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin Patrick Murphy, and Tim Salimans. Diffusion models and gaussian flow matching: Two sides of the same coin. In *The Fourth Blogpost Track at ICLR 2025*, 2025b.

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. URL https://arxiv.org/abs/2505.13447.

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

Eric Goldman. An introduction to the california consumer privacy act (ccpa). *Santa Clara Univ. Legal Studies Research Paper*, 2020.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Nikita Gushchin, David Li, Daniil Selikhanovych, Evgeny Burnaev, Dmitry Baranchuk, and Alexander Korotin. Inverse bridge matching distillation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21471–21496. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gushchin25b.html.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://arxiv.org/abs/2006.11239.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky TQ Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. *arXiv preprint arXiv:2410.20587*, 2024.

Wan Jiang, He Wang, Xin Zhang, Dan Guo, Zhaoxin Fan, Yunfeng Diao, and Richang Hong. Moderating the generalization of score-based generative model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 360–369, October 2025.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24174–24184. IEEE, 2024.

Shervin Khalafi, Alejandro Ribeiro, and Dongsheng Ding. Unlearning in diffusion models: A unified framework with KL divergence and likelihood constraints. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.30825.

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. URL https://openreview.net/forum?id=ymjI8feDTD.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Nikita Kornilov, David Li, Tikhon Mavrin, Aleksei Leonov, Nikita Gushchin, Evgeny Burnaev, Iaroslav Koshelev, and Aleksandr Korotin. Universal inverse distillation for matching models with real-data supervision (no gans). In *International Conference on Learning Representations*, volume 2026, pp. 67906–67948, 2026.

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*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Ping Liu and Chi Zhang. Erased or dormant? rethinking concept erasure through reversibility. *arXiv preprint arXiv:2505.16174*, 2025.

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. URL https://openreview.net/forum?id=XVjTT1nw5z.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Kevin Lu, Nicky Kriplani, Rohit Gandikota, Minh Pham, David Bau, Chinmay Hegde, and Niv Cohen. When are concepts erased from diffusion models? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 14751–14775. Curran Associates, Inc., 2025. doi: 10.52202/085713-0497. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/15bbe6ddfc88d8e7f59c8f7d4e2541f5-Paper-Conference.pdf.

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

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):1–46, 2025.

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*, 2024.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11410–11420, June 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Aakash Sen Sharma, Niladri Sarkar, Vikram Chundawat, Ankur A Mali, and Murari Mandal. Unlearning or concealment? a critical analysis and evaluation metrics for unlearning in diffusion models. *arXiv preprint arXiv:2409.05668*, 2024.

Qitan Shi, Cheng Jin, Jiawei Zhang, and Yuantao Gu. Retrack: Data unlearning in diffusion models through redirecting the denoising trajectory. In Emtiyaz Khan, Yingzhen Li, Arno Solin, and Aaditya Ramdas (eds.), *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 2818–2826. PMLR, 02–05 May 2026. URL https://proceedings.mlr.press/v300/shi26a.html.

Lorenzo Simone, Davide Bacciu, and Shuangge Ma. Continualflow: Learning and unlearning with neural flow matching. In *ICML 2025 Workshop on Machine Unlearning for Generative AI (MUGen)*, 2025. URL https://arxiv.org/abs/2506.18747.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

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*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Haoran Tang and Rajiv Khanna. Sharpness-aware machine unlearning. In *International Conference on Learning Representations*, volume 2026, pp. 79941–79985, 2026.

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.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr): a practical guide. 2017.

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

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024b.

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

Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62307–62331. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhou24x.html.

CONTENTS

1 Introduction **1**

1.1 Contributions 2

2 Related Work **2**

2.1 Diffusion, flow and matching models 2

2.2 Distillation and one-step models 3

2.3 Data unlearning 3

2.4 Class unlearning 4

3 Inverse Distillation Unlearning **4**

3.1 Method Description 4

3.2 Practical details 5

4 Experiments **6**

5 Discussion and comparison **8**

A Proofs and related methods **15**

A.1 Proof of Theorem 1 15

A.2 The minimized distance 15

A.3 SFD's details 16

B Experimental details **17**

B.1 Architectures and teacher checkpoints 17

B.2 Teacher sampling 17

B.3 Evaluation classifiers 17

B.4 FID protocol and evaluation data 17

B.5 Optimization hyperparameters 18

B.6 Initialization 18

B.7 Fine-tuning experiments 18

B.8 Code and checkpoint release 18

B.9 MNIST FM retraining and sensitivity to ρ 18

C Visual Results **20**

A PROOFS AND RELATED METHODS

A.1 PROOF OF THEOREM 1

First, we need the formal convergence guarantees for the inverse distillation scheme (5).

Lemma 1 (Inverse distillation scheme's optimum (Kornilov et al., 2026)) *The inverse scheme (5) with the teacher $f^* = \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^*)$ always attains its optimum 0 when and only when the teacher data is retrieved, i.e., $p_0 = p_0^*$.*

Proof. According to Lemma 1, after we optimize the inverse distillation scheme (5) over distribution p_0, we get the teacher data p_0^* distilled into this optimized distribution, i.e., $p_0 = p_0^*$. Since we parametrize the optimized distribution as a mixture of the generated and forget data $p_0 = \rho\, p_0^F + (1-\rho)\, p_0^\theta$ with $\rho = \pi$, then at the optimal parameters θ_{opt}, we get:

$$p_0 = \rho\, p_0^F + (1-\rho)\, p_0^{\theta_{\text{opt}}} = \pi\, p_0^F + (1-\pi)\, p_0^{\theta_{\text{opt}}} = p_0^*. \tag{8}$$

Finally, considering the structure of the teacher data (4), we conclude that:

$$\pi\, p_0^F + (1-\pi)\, p_0^{\theta_{\text{opt}}} \overset{(8)}{=} p_0^* \overset{(4)}{=} \pi\, p_0^F + (1-\pi)\, p_0^R \quad \Rightarrow \quad p_0^{\theta_{\text{opt}}} = p_0^R.$$

□

A.2 THE MINIMIZED DISTANCE

First, we use the formula of the IDU loss (6) with the optimal fake model $f^{\text{mix}} := \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^{\text{mix}})$ on the current mixed data $p_0^{\text{mix}} := \rho \cdot p_0^F + (1-\rho) \cdot p_0^\theta$:

$$\mathcal{L}_{\text{IDU}}(f^{\text{mix}}, p_0^\theta) = \mathcal{L}_{\text{UM}}(f^*, p_0^{\text{mix}}) - \mathcal{L}_{\text{UM}}(f^{\text{mix}}, p_0^{\text{mix}}).$$

Following the structure of UM loss (1) on the mixed data, we get:

$$\begin{aligned}
\mathcal{L}_{\text{UM}}(f, p_0^{\text{mix}}) &= \mathbb{E}_{t,\, x_0 \sim p_0^{\text{mix}},\, x_t \sim p_t^{\text{mix}}(\cdot|x_0)} \left[\left\| f_t(x_t) - f_t^{\text{mix}}(x_t \mid x_0) \right\|^2 \right] \\
&= \mathbb{E}_{t,\, x_t \sim p_t^{\text{mix}},\, x_0 \sim p_0^{\text{mix}}(\cdot|x_t)} \left[\left\| f_t(x_t) - f_t^{\text{mix}}(x_t \mid x_0) \right\|^2 \right] \\
&= \mathbb{E}_{t,\, x_t \sim p_t^{\text{mix}}} \left[\left\| f_t(x_t) - \mathbb{E}_{x_0 \sim p_0^{\text{mix}}(\cdot|x_t)} [f_t^{\text{mix}}(x_t \mid x_0)] \right\|^2 \right] + C(p_0^{\text{mix}}), \\
C(p_0^{\text{mix}}) &= \mathbb{E}_{t,\, x_t \sim p_t^{\text{mix}}} \left[\mathbb{E}_{x_0 \sim p_0^{\text{mix}}(\cdot|x_t)} \left[\left\| f_t^{\text{mix}}(x_t \mid x_0) \right\|^2 \right] \right] \\
&- \mathbb{E}_{t,\, x_t \sim p_t^{\text{mix}}} \left[\left\| \mathbb{E}_{x_0 \sim p_0^{\text{mix}}(\cdot|x_t)} \left[f_t^{\text{mix}}(x_t \mid x_0) \right] \right\|^2 \right],
\end{aligned}$$

where $C(p_0^{\text{mix}})$ is the bias–variance decomposition term independent of f. For fixed t, x_t, the UM loss is minimized by the conditional mean:

$$\begin{aligned}
f_t^{\text{mix}}(x_t) &= \mathbb{E}_{x_0 \sim p_0^{\text{mix}}(\cdot|x_t)} [f_t^{\text{mix}}(x_t \mid x_0)] = \arg\min_f \mathcal{L}_{\text{UM}}(f, p_0^{\text{mix}}), \\
\mathcal{L}_{\text{UM}}(f, p_0^{\text{mix}}) &= \mathbb{E}_{t,\, x_t \sim p_t^{\text{mix}}} \left[\left\| f_t(x_t) - f_t^{\text{mix}}(x_t) \right\|^2 \right] + C(p_0^{\text{mix}}).
\end{aligned}$$

Thus, for the IDU loss with the optimal fake model f^{mix}, we have:

$$\begin{aligned}
\mathcal{L}_{\text{IDU}}(f^{\text{mix}}, p_0^\theta) &= \mathcal{L}_{\text{UM}}(f^*, p_0^{\text{mix}}) - \mathcal{L}_{\text{UM}}(f^{\text{mix}}, p_0^{\text{mix}}) \\
&= \mathbb{E}_{t,\, x_t \sim p_t^{\text{mix}}} \left[\left\| f_t^*(x_t) - f_t^{\text{mix}}(x_t) \right\|^2 \right] + C(p_0^{\text{mix}}) \\
&- \mathbb{E}_{t,\, x_t \sim p_t^{\text{mix}}} \left[\left\| f_t^{\text{mix}}(x_t) - f_t^{\text{mix}}(x_t) \right\|^2 \right] - C(p_0^{\text{mix}}) \\
&= \mathbb{E}_{t,\, x_t \sim p_t^{\text{mix}}} \left[\left\| f_t^*(x_t) - f_t^{\text{mix}}(x_t) \right\|^2 \right].
\end{aligned}$$

A.3 SFD'S DETAILS

SFD (Chen et al., 2025) distills a conditional teacher model into a one-step generator in parallel with class forgetting. It splits the generator loss from the inverse distillation scheme (2) into the forget and remaining losses as in (3): in the forget loss, it aligns the conditional scores of the generator's forget classes c_F with the teacher's scores for the safe classes c_S; in the remaining loss, it leaves other classes c_R unswapped. The method also modifies these losses for better convergence, following the SiD framework (Zhou et al., 2024):

$$\begin{aligned}\mathcal{L}_{\text{SFD-gen-remain}}(p_0^\theta) &= \mathbb{E}_{t,x_0^\theta\sim p_0^\theta(\cdot|c_R),x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta,c_R)}[-2\cdot\alpha_{\text{SiD}}\|f_t^*(x_t^\theta|c_R)-f_t(x_t^\theta|c_R)\|^2 \\ &+ 2\langle f_t^*(x_t^\theta|c_R)-f_t(x_t^\theta|c_R), f_t^*(x_t^\theta|c_R)-f_t^\theta(x_t^\theta|x_0^\theta,c_R)\rangle], \quad (9)\\ \mathcal{L}_{\text{SFD-gen-forget}}(p_0^\theta) &= \mathbb{E}_{t,x_0^\theta\sim p_0^\theta(\cdot|c_F),x_t^\theta\sim p_t^\theta(\cdot|x_0^\theta,c_F)}[-2\cdot\alpha_{\text{SiD}}\|f_t^*(x_t^\theta|c_S)-f_t(x_t^\theta|c_F)\|^2 \\ &+ 2\langle f_t^*(x_t^\theta|c_S)-f_t(x_t^\theta|c_F), f_t^*(x_t^\theta|c_S)-f_t^\theta(x_t^\theta|x_0^\theta,c_F)\rangle], \quad (10)\end{aligned}$$

where α_{SiD} is an arbitrary parameter, usually taken from the range $[0.5, 1.2]$. The loss for the fake model remains the same for all classes.

B EXPERIMENTAL DETAILS

B.1 ARCHITECTURES AND TEACHER CHECKPOINTS

For CIFAR-10, the FM setup uses the TorchCFM U-Net architecture from the public 400k-step Independent Conditional Flow Matching (I-CFM) checkpoint[1], whereas the SiD setup uses the DDPM++ (SongUNet) architecture of the public unconditional EDM-VP teacher adopted by the official SiD implementation[2]. Both CIFAR-10 teachers operate on 32×32 RGB images. The public checkpoints are `cfm_cifar10_weights_step_400000.pt` for FM and `edm-cifar10-32x32-uncond-vp.pkl` for SiD.

For MNIST, we preserve each architecture family but adapt it to 28×28 grayscale inputs. In the FM U-Net, we change the input and output channels from 3 to 1, reduce the resolution hierarchy from `[1,2,2,2]` to `[1,2,2]`, move self-attention from resolution 16 to 14, reduce the number of channels per attention head from 64 to 32. In the SiD DDPM++ model, we change only the image resolution from 32 to 28, the image channels from 3 to 1, and the attention resolution from 16 to 14.

B.2 TEACHER SAMPLING

For the CIFAR-10 FM teacher, we use the adaptive Dormand–Prince (Dopri5) ODE solver with relative and absolute tolerances of 10^{-5}, following the original TorchCFM setup. For the MNIST FM teacher, we use a fixed-step Euler solver with 100 integration steps. For both the MNIST and CIFAR-10 SiD teachers, we use the deterministic 18-step EDM sampler with the second-order correction prescribed by the original EDM and SiD implementations. Every distilled generator produces a sample in one step.

B.3 EVALUATION CLASSIFIERS

To compute FGR, we classify 50,000 generated images and report the percentage assigned to each forgotten class. For MNIST, we use the public LeNet-5 checkpoint[3]; its repository reports 99.13% validation accuracy and 98.94% test accuracy. For CIFAR-10, we use the public ResNet-56 checkpoint[4], for which the repository reports 94.37% top-1 and 99.83% top-5 accuracy. These classifiers are external measurement instruments: they are not part of IDU, are never queried by the training loop, and contribute no loss, gradient, feature representation, or conditioning signal.

B.4 FID PROTOCOL AND EVALUATION DATA

All reported FID values use `clean-fid` v0.1.35 (Parmar et al., 2022) in `legacy_tensorflow` mode and 50,000 generated images. Images are quantized before extracting 2048-dimensional TensorFlow-compatible Inception features; grayscale MNIST samples are replicated across three channels. We use all available real training images from the distribution targeted by each row: the full training set for the full-data teacher and pure distillation, and the corresponding training subset with the forgotten labels removed for IDU and retained-only retraining. Thus, the paired-class references contain 60,000 full or 47,604 retained MNIST images and 50,000 full or 40,000 retained CIFAR-10 images. Single-class experiments analogously use the complete training subset excluding the selected class. Test images are not used as FID references.

Under the same legacy feature pipeline, the only numerical convention that differs from the original SiD evaluator is covariance normalization: CleanFID uses the sample covariance denominator $N-1$, whereas the original SiD/StyleGAN implementation uses the population denominator N. At 50,000 samples this changes the covariance scale only by the factor $N/(N-1) \approx 1.00002$. Minor implementation-level differences remain in covariance symmetrization and numerical stabilization. Teacher FID uses the multi-step samplers described above; every distilled or IDU generator is evaluated with one-step inference.

[1] https://github.com/atong01/conditional-flow-matching/tree/main/examples/images/cifar10

[2] https://github.com/mingyuanzhou/SiD

[3] https://github.com/hrfang/LeNet5-code-examples

[4] https://github.com/chenyaofo/pytorch-cifar-models

B.5 Optimization hyperparameters

FM on MNIST. Full-data and retained-only teachers use learning rate 10^{-4}, total batch size 128, and 100k and 50k optimizer steps, respectively. The retained-only teacher required fewer iterations because it empirically converged faster, likely due to the smaller amount of training data. Pure distillation, retained-only distillation, and IDU use learning rate 10^{-4}, total batch size 256, and training horizons of approximately 30,000 generator iterations. Teacher pretraining uses Adam with $\beta = (0.9, 0.999)$; distillation and IDU use Adam with $\beta = (0, 0.999)$ for both trainable networks. All MNIST FM optimizers use cosine learning-rate annealing over their respective horizons.

FM on CIFAR-10. The 400k-step teacher recipe uses learning rate 2×10^{-4}, global batch size 128, 5,000 warm-up steps, and 400,000 optimizer updates. Pure and retained-only distillation use learning rate 3×10^{-5}, total batch size 256, and approximately 50,000 generator iterations; IDU uses the same learning rate and horizon with configured per-process batch size 256. These runs use 500 warm-up steps. Across FM distillation and IDU runs, we use gradient clipping at 1, generator EMA 0.999, and $\alpha_{\mathrm{SiD}} = 0.5$; the CIFAR-10 teacher itself uses EMA 0.9999.

SiD experiments. MNIST teacher training uses learning rate 10^{-4}, global batch size 512, and a 64-million-image (64 Mimg) horizon. The CIFAR-10 teacher recipe uses learning rate 10^{-3}, global batch size 512, and a 200 Mimg training horizon; the full-data result uses the public EDM-VP checkpoint, while retained-only teachers follow this recipe. On both datasets, IDU, pure, and retained-only distillation use learning rates 10^{-5} for the generator and fake score network, global batch size 512, and a 100 Mimg training horizon. We retain $\alpha_{\mathrm{SiD}} = 1.2$, $t_{\max} = 800$, and initial noise standard deviation 2.5 from the original SiD recipe (Zhou et al., 2024).

These values are maximum optimization horizons rather than a claim that later checkpoints are always preferable. We select the reported checkpoint by the lowest observed FID; for FM, selection additionally precedes the late forgetting reversal discussed in Section 5.

B.6 Initialization

Our IDU can be used for both distillation from scratch and fine-tuning of an already pretrained generator. The only difference between the setups, besides the hyperparameter values, is the initialization: for fine-tuning, we initialize G_θ from the pretrained generator; otherwise, we initialize it from the one-step teacher inference scheme.

B.7 Fine-tuning experiments

We evaluate fine-tuning on CIFAR-10 by initializing from the corresponding pure-distillation checkpoint and continuing IDU optimization with learning rates 5×10^{-6} for FM and 10^{-6} for SiD, retaining the forgetting weights from the main experiments: $\rho = 0.6$ for FM and $\rho = 0.2$ for SiD. For forgotten classes 1 and 9, FM fine-tuning yields Retain FID 5.17 ± 0.08 and FGRs $0.89 \pm 0.02\%$ and $0.42 \pm 0.03\%$, respectively. SiD fine-tuning yields Retain FID 3.12 ± 0.06 and FGRs $2.69 \pm 0.06\%$ and $2.60 \pm 0.04\%$; see Table 4 for comparison. Both fine-tuned models reduce forgotten-class generation relative to pure distillation, but the trade-off differs by backbone: compared with IDU trained from scratch, FM improves Retain FID while slightly increasing FGR, whereas SiD maintains comparable Retain FID but higher FGR.

B.8 Code and checkpoint release

Upon publication, we will release the complete source code, exact configurations, evaluation scripts, and checkpoints used to produce the main reported results.

B.9 MNIST FM retraining and sensitivity to ρ

The retained-only FM baseline on MNIST exhibits a distinct sensitivity. The exceptionally low Retain FID of the retrained teacher (0.34) may reflect strong overfitting to the smaller retained subset rather than uniformly better generalization. In its distilled counterpart, we consistently observed poor generation of digit 2, which raises Retain FID to 5.72. This failure occurred despite using

Table 4: Fine-tune FM/SiD results for the purely distilled generators on CIFAR-10. FID uses full-data references for Pretrain/Pure distillation and retained-data references for IDU and fine-tuning; FGR is reported per forgotten class. Values are mean ± standard deviation over five runs, except † values from (Zhou et al., 2024).

	FM		SiD	
Mode	FID ↓	FGR (%) ↓ Class 1 / Class 9	FID ↓	FGR (%) ↓ Class 1 / Class 9
Pretrain	3.66 ± 0.03	12.50 ± 0.08 11.18 ± 0.12	$1.97^{\dagger}$	11.13 ± 0.12 10.04 ± 0.15
Pure distillation	4.35 ± 0.05	7.58 ± 0.09 8.75 ± 0.21	$1.92 \pm 0.02^{\dagger}$	10.11 ± 0.14 10.66 ± 0.14
Forgotten classes	$\{1, 9\}$		$\{1, 9\}$	
IDU from scratch ($\rho_{\text{CIFAR-10}} = 0.6$) ($\rho_{\text{SiD}} = 0.2$)	5.81 ± 0.05	0.56 ± 0.05 0.39 ± 0.03	3.15 ± 0.03	1.11 ± 0.06 1.25 ± 0.06
IDU Fine-tuning ($\rho_{\text{CIFAR-10}} = 0.6$) ($\rho_{\text{SiD}} = 0.2$)	5.17 ± 0.08	0.89 ± 0.02 0.42 ± 0.03	3.12 ± 0.06	2.69 ± 0.06 2.60 ± 0.04

Table 5: Effect of the forget-mixture weight ρ on FM-based IDU for MNIST when jointly forgetting digits 3 and 7. The main configuration is bold. The final column lists qualitatively suppressed digits in addition to the target digits 3 and 7.

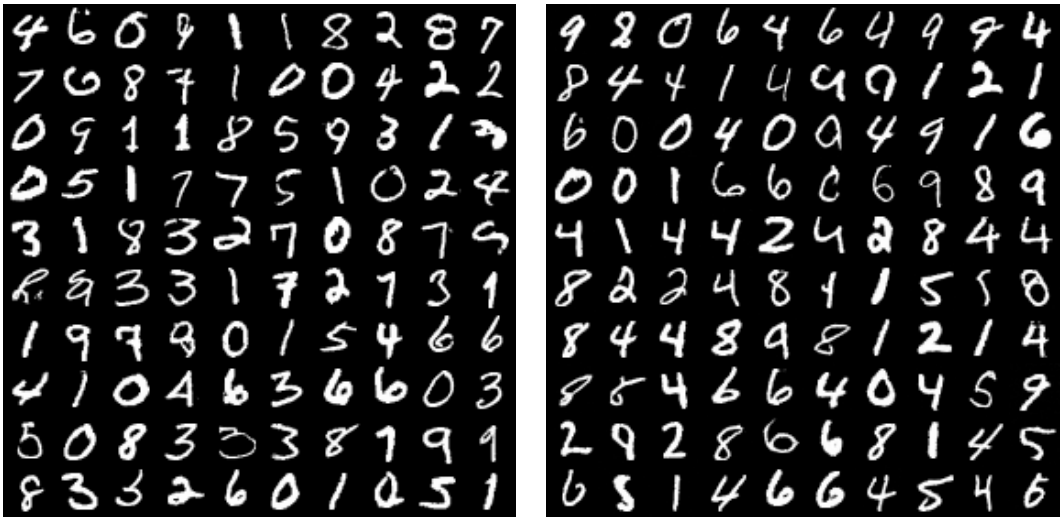
ρ	Retain FID ↓	FGR 3 (%) ↓	FGR 7 (%) ↓	Additional suppressed digits
0.05	4.29 ± 0.02	7.07 ± 0.06	8.18 ± 0.06	—
0.1	3.96 ± 0.04	3.87 ± 0.09	4.62 ± 0.10	—
0.2	4.81 ± 0.08	2.22 ± 0.07	1.63 ± 0.07	—
0.4	$\mathbf{3.57 \pm 0.03}$	$\mathbf{0.16 \pm 0.01}$	$\mathbf{0.16 \pm 0.01}$	—
0.6	11.20 ± 0.10	0	0	2, 5, 8
0.8	10.58 ± 0.07	0	0	2, 5
0.9	10.35 ± 0.04	0	0	2, 5, 9

the same architecture and optimization settings as the stable full-data pretraining and distillation runs, suggesting sensitivity of the retraining baseline to the altered data distribution rather than an intentional hyperparameter disadvantage.

Table 5 reports the FM/MNIST results for different ρ. The main setting, $\rho = 0.4$, gives the best observed Retain FID–FGR trade-off. Increasing ρ to 0.6 or above drives the target-class FGR values to zero, but also suppresses additional digits—most consistently 2 and 5, with 8 or 9 affected in some runs—and raises Retain FID above 10. Conversely, decreasing ρ below 0.4 progressively weakens forgetting of digits 3 and 7 and does not improve Retain FID over the main setting. This sensitivity appears specific to the low-dimensional MNIST setting and the FM architecture and training recipe used here, and motivates evaluation with larger and more stable backbones. In contrast, the more recent SiD distillation backbone remained stable at $\rho = 0.2$, which matches the nominal fraction of two forgotten classes among ten, without the same collateral class suppression.

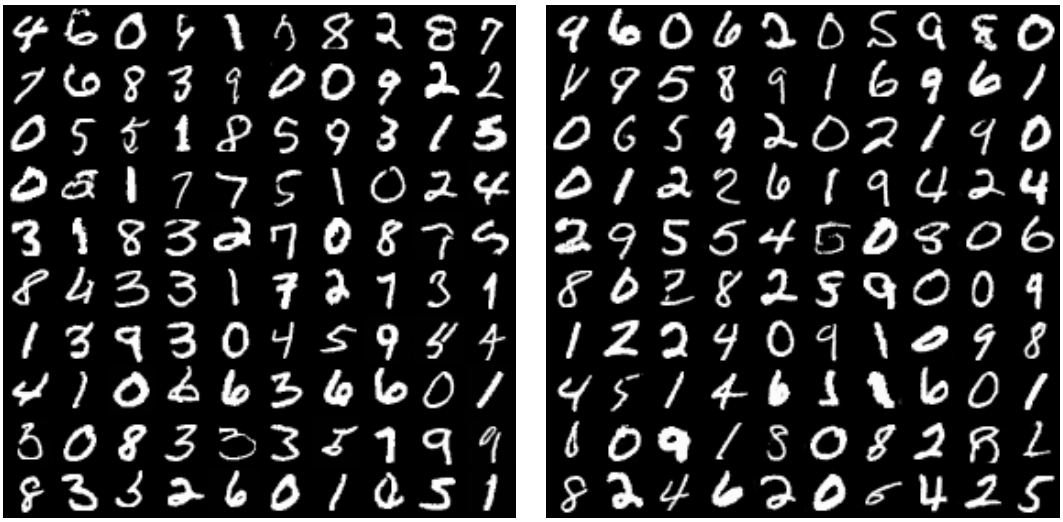
C VISUAL RESULTS

We qualitatively compare samples from each multi-step teacher with samples from the corresponding one-step IDU generator. For MNIST, IDU is trained to forget digits 3 and 7; for CIFAR-10, it is trained to forget automobile (class 1) and truck (class 9). Across both FM and SiD, the teacher grids contain the target classes, whereas the IDU grids visibly suppress them while preserving samples from the retained classes. These finite grids are intended as qualitative illustrations; the corresponding 50,000-sample FGR measurements are reported in Table 1.



Teacher IDU

Figure 2: FM on MNIST: teacher and IDU samples when forgetting digits 3 and 7.



Teacher IDU

Figure 3: SiD on MNIST: teacher and IDU samples when forgetting digits 3 and 7.



Teacher IDU

Figure 4: FM on CIFAR-10: teacher and IDU samples when forgetting automobile (class 1) and truck (class 9).



Teacher IDU

Figure 5: SiD on CIFAR-10: teacher and IDU samples when forgetting automobile (class 1) and truck (class 9).