Crossing Lethe: Steering Probability Flows for Data Unlearning

Anonymous authors
Paper under double-blind review

Abstract

Data unlearning (the targeted removal of specific training examples from a deployed model) is increasingly important for privacy compliance, user rights, and safe model updating. While unlearning has been actively studied for energy-based models and diffusion models, comparatively little work addresses flow matching models, in part because it is unclear how removing data should alter the learned dynamics. Flow matching trains continuous generative models by learning a time-dependent velocity field whose estimates aggregate contributions from all training samples, making post-training selective removal nontrivial. We propose a flow decomposition framework that adapts to the level of data access and operational constraints. We first introduce Loss-level Flow Decomposition (LFD), that explicitly subtracting the forget-set contribution. Importantly, LFD serves as a global optimizer that obviates the need to retrain the model. LFD provides high fidelity in standard settings where both retain and forget datasets are available. We then investigate privacy-restricted scenarios where explicit access to forget samples is prohibited or governed by strict constraints. Specifically, we introduce an Energy-Based Reformulation (EBR) that leverages a proxy classifier rather than direct forget data samples. We rigorously analyze both these approaches. We also validate our framework on several datasets, demonstrating that these methods effectively balance unlearning performance with the preservation of model fidelity across both standard and restricted data regimes.

1 Introduction

The ability to remove the influence of specific training data from deployed machine learning models has become a pressing requirement in modern AI systems (Cao & Yang, 2015; Ginart et al., 2019; Bourtoule et al., 2021). Such requests arise from data-governance regimes (Regulation, 2016; Court of Justice of the European Union), dataset curation errors, and the need for safe and continual model updates, and they demand methods that selectively erase designated data *without* the prohibitive cost of retraining from scratch. Because flow-matching models generate by transporting probability along continuous *flows*, we name our approach after *Lethe*, the mythological river of forgetting: unlearning here amounts to steering those flows away from the data marked for removal. Throughout, we study *empirical, output-level* removal, which is distinct from a certified privacy or membership-removal guarantee.

While data unlearning has been actively studied for discriminative models, energy-based models, and diffusion-based generative models (Guo et al., 2020; Golatkar et al., 2020; Bourtoule et al., 2021; Alberti et al., 2025; Zhang et al., 2024; Kumari et al., 2023; Gandikota et al., 2023), comparatively little work addresses unlearning in flow matching models (Simone et al., 2025; Gao et al., 2025; Fan et al., 2026), despite its growing adoption as a flexible and efficient framework for continuous generative modeling (Lipman et al., 2023; Liu et al., 2022; Albergo et al., 2023; Esser et al., 2024; Holderrieth & Erives, 2025). The difficulty is intrinsic to the model class: flow matching trains a velocity field by regressing on samples drawn from the *entire* data distribution, so the influence of any one forget point is entangled across the learned dynamics rather than localized. Existing per-sample removal recipes therefore do not transfer directly, and naively retraining on the retained data alone is clean but often prohibitively expensive. We study data unlearning in flow matching from a unifying *objective-level* perspective: rather than an ad hoc post-processing step, unlearning becomes a principled modification of the flow-matching training objective. Our central requirement

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.

is that, after unlearning, the model approximate one trained *only* on the retained data, the exact-retraining reference, so that retained-data generation is preserved while the forget set is suppressed. Which modification is appropriate depends on what is known about the data to remove. When the retain–forget split is explicit, we introduce **Loss-level Flow Decomposition (LFD)**, which removes the forget contribution directly from the flow-matching objective and yields a stable estimator of the retain-only training loss. When forgetting is instead specified only through a proxy signal (a classifier or detector scoring how likely a sample is forget-associated), we introduce **Energy-Based Reweighting (EBR)**, which uses that signal to softly suppress forget-like samples without converting uncertain membership into a hard partition.

We validate each method in its natural access regime: LFD on precise identity removal in CelebA and EBR on digit-class removal in MNIST. LFD drives target-identity similarity down to the exact-retraining reference while leaving retained-set image quality essentially unchanged, and EBR removes an entire digit class using only a proxy signal. Both operate as short fine-tunes: unlearning takes tens to a few hundred update steps (rather than the tens of thousands used to retrain from scratch) at a single model evaluation per step.

1.1 Our Contributions

Contribution	Broad claim	Where addressed
Framework	Cast unlearning in flow matching as *objective-level* retain-risk estimation: approximate the exact-retraining reference by modifying the flow-matching objective.	§ 4
LFD	With a known retain–forget split, decompose the flow loss to remove the forget contribution exactly, yielding an unbiased single-pass estimator of the retain-only loss.	§ 5 Thm. 5.1, 5.3
EBR	With only a proxy signal, reweight the endpoint law by a retain-posterior energy that softly suppresses forget-like samples, with exact oracle recovery and a proxy-insufficiency bound.	§ 6 Thm. 6.1, 6.2; Prop. B.3
Experiments	Validate each method in its natural regime: LFD matches the exact-retraining reference for CelebA identity removal while preserving image quality; EBR removes an entire MNIST digit class from proxy signal alone.	§ 7 § 7.1, 7.2

2 Related Work

Unlearning. Machine unlearning studies how to remove the influence of a designated forget set from a trained model, motivated by privacy regulation, user rights, and continual updates. Prior work spans retraining-based baselines, certified removal guarantees, and efficient approximate procedures (e.g., SISA-style sharding/slicing) (Bourtoule et al., 2021). Two recent surveys systematize the space of definitions, threat models, evaluation protocols, and attacks against weak unlearning (Xue et al., 2026; Liu et al., 2025). Because empirical evaluation is subtle and frequently confounded by mismatched threat models and metrics, benchmark-driven comparisons have become increasingly common (Cadet et al., 2025).

Unlearning in diffusion models. Diffusion models (Ho et al., 2020; Song et al., 2020) have become a primary testbed for unlearning in modern generative modeling, leading to a large body of methods that (i) fine-tune to reduce forget influence while preserving utility on retained data, (ii) cast unlearning as constrained or bi-level optimization, or (iii) target concept- or instance-level erasure through editing objectives. Representative examples include saliency/weight-attribution based unlearning (Fan et al., 2023), constrained/trajectory-deviation style procedures (Wu et al., 2025), text-to-image instance forgetting (Zhang et al., 2024), and concept erasure in text-conditioned diffusion (Gandikota et al., 2023). Closest in spirit to our estimator viewpoint are approaches that explicitly decompose or subtract forget contributions to recover a retain-equivalent training signal (Alberti et al., 2025). We focus on diffusion models as a close cousin to flow matching: both learn time-dependent vector

fields along stochastic bridges, and diffusion can be viewed as a particular noisy path family within this broader framework (Lipman et al., 2023).

Unlearning in flow matching models. Flow matching learns a velocity field that aggregates per-example contributions over a time-indexed bridge (Lipman et al., 2023), making post-hoc removal nontrivial. To date, the only work explicitly addressing unlearning in neural flow matching is *ContinualFlow* (Simone et al., 2025), which proposes constructing a new transport model that maps samples from a pretrained generator distribution q_0 to a retained distribution q_1 using an OT-CFM coupling argument (cf. the OT-CFM justification attributed to Tong et al.) (Tong et al., 2023). This yields a *two-model* deployment in which the original generator (trained on retain+forget) remains intact and is used to produce inputs that are subsequently transformed toward the retained distribution. Its public implementation releases only a low-dimensional 2D baseline, so any image-domain comparison necessarily reimplements its energy-reweighting rule rather than reproducing the published system. We instead aim to release a *single deployed generator* whose own parameters realize the unlearning, rather than a two-model pipeline that leaves the original retain+forget generator intact behind a post-hoc transformation. This is a *deployment* desideratum (we ship one sanitized model rather than relying on withholding the original generator) and not a certified privacy or weight-level guarantee, consistent with the empirical, output-level scope we adopt throughout. In this sense a post-hoc transformation layer does not by itself sanitize the underlying model, whose retain+forget parameters remain unchanged.

The closest estimator-based approach is SISS (Alberti et al., 2025), which subtracts the forget contribution by importance sampling for *diffusion* (score-based) models; in the stochastic limit our loss-level decomposition reduces to it (Appendix E). Our contribution is to extend this retain-risk decomposition to *deterministic* flow matching (the velocity parameterization of continuous flows) rather than to introduce the decomposition idea itself.

3 Preliminaries

Flow matching. Generative modeling aims to simulate samples from an unknown data distribution $P_{\text{data}}(z)$, which typically admits neither a closed-form density nor an easy sampler. Flow-based models instead learn a transformation from a simple source $P_0(x)$ (e.g., a standard Gaussian) to the data: a continuous, time-indexed map whose marginal $P_t(x)$ evolves from P_0 at $t = 0$ to $P_1 = P_{\text{data}}$ at $t = 1$. Flow matching realizes this through an ordinary differential equation (ODE)

$$\frac{dx}{dt} = u_t(x),$$

with a time-dependent velocity field $u_t(x)$; solving it from $t = 0$ to $t = 1$ pushes P_0 forward to P_{data}, so the central learning problem is to estimate $u_t(x)$.

Rather than fitting u_t indirectly via likelihood maximization, flow matching constructs its target explicitly through *bridge distributions* $P_t(x \mid z)$ that interpolate between noise at $t = 0$ and a data point $z \sim P_{\text{data}}$ at $t = 1$, each carrying a conditional velocity $u_t(x \mid z)$. The population target aggregates these by posterior responsibility,

$$u_t^{\text{target}}(x) = \int \frac{P_{\text{data}}(z)\, P_t(x \mid z)}{P_t(x)}\, u_t(x \mid z)\, dz, \qquad P_t(x) = \int P_{\text{data}}(z)\, P_t(x \mid z)\, dz.$$

Fitting this field directly yields the *flow matching loss* ℓ_{FM}, which is intractable because it requires expectations over the marginal $P_t(x)$. Flow matching therefore optimizes the *conditional* variant: sample $z \sim P_{\text{data}}$, $t \sim \text{Unif}[0, 1]$, and $x \sim P_t(\cdot \mid z)$, then regress the learned field $u_t^\theta(x)$ onto the conditional target $u_t(x \mid z)$. This *conditional flow matching* (CFM) loss ℓ_{CFM} differs from ℓ_{FM} only by a θ-independent constant, so the two share the same minimizer and expected gradients while ℓ_{CFM} is tractable. We adopt the CFM formulation throughout.

Data unlearning. Data unlearning aims to remove the influence of a designated subset of training data from a deployed model without full retraining. Let P_{data} denote the full data distribution and associate each sample z with an indicator $R \in \{0, 1\}$, where $R = 1$ marks *retain* data and $R = 0$ *forget* data. Writing $\pi_R = \Pr(R = 1)$ and $\pi_F = \Pr(R = 0) = 1 - \pi_R$ induces the conditionals $P_{\text{retain}} \equiv P_{\text{data}}(z \mid R = 1)$ and $P_{\text{forget}} \equiv P_{\text{data}}(z \mid R = 0)$, so that $P_{\text{data}} = \pi_R\, P_{\text{retain}} + \pi_F\, P_{\text{forget}}$.

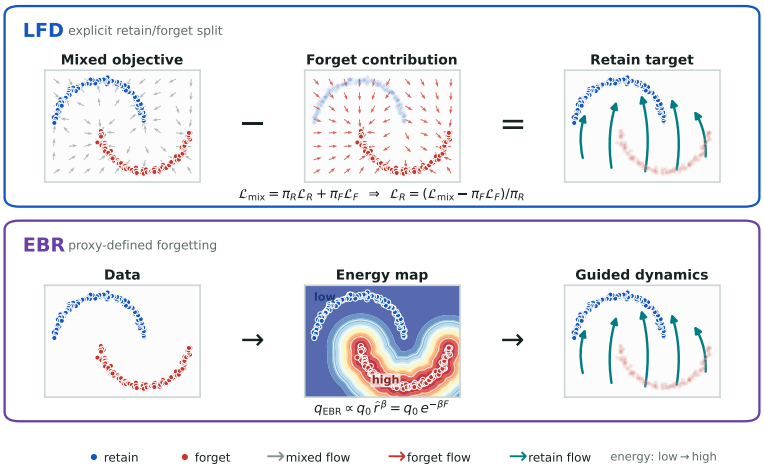

Figure 1: Illustration of LFD and EBR on a two-moons unlearning task. LFD removes the forget contribution at the objective level to recover a retain-only target, whereas EBR assigns high energy to forget regions and guides the learned dynamics toward low-energy retain regions.

Unlearning seeks a model whose behavior matches training on P_{retain} alone. For flow matching this is nontrivial: the velocity field $u_t(x)$ aggregates contributions from all training samples, entangling retain and forget information in the dynamics. Unlearning therefore requires selectively removing the contribution of P_{forget} while preserving that induced by P_{retain}, and doing so without full retraining, by decomposing or reweighting the learned dynamics with respect to the retain and forget distributions.

4 Our Approach

We give a unified perspective on data unlearning in flow matching that leads to both methods proposed in this work: reason directly about how probability mass moves under the learned dynamics, and how that motion must be altered to remove the influence of forgotten data.

Flow dynamics and probability mass. A flow-matching model transports probability mass from a simple source to the data distribution via a time-dependent velocity field $u_t(x)$, and the induced density $P_t(x)$ evolves by the continuity equation

$$\partial_t P_t(x) + \nabla \cdot \big(P_t(x)\, u_t(x)\big) = 0,$$

a conservation principle: mass is neither created nor destroyed, only redistributed by the field, with the local divergence $\nabla \cdot u_t(x)$ indicating whether mass accumulates or disperses near x (the exact rate also includes the transport term $u_t \cdot \nabla P_t$). Trained on the full dataset, the field drives mass toward every training point, so at $t = 1$ the density concentrates there; near a forget point it therefore acts as an *attractor*, pulling nearby trajectories inward. Unlearning must suppress this training signal so that forget samples no longer shape the learned dynamics in the same way, and how to carry out that suppression depends on what information is available about the retain and forget sets.

Two routes to suppressing forget influence. We consider two complementary objective-level strategies, each suited to a different information regime.

Explicit retain–forget split: Loss-level Flow Decomposition (LFD). When both sets are explicitly available, the most direct route is to modify the flow-matching loss itself: since training minimizes an expected regression loss over data samples, we can construct an estimator whose expectation matches the retain-only objective. LFD removes the forget contribution at the loss level and thereby approximates the model obtained by retraining on retained data alone. It is the natural choice in the fully specified setting, realizing the removal in the deployed model's own parameters (stable *weight-level* unlearning) rather than a post-hoc transformation.

Soft or uncertain forget membership: Energy-Based Reweighting (EBR). Often the forget set is not a clean hard partition but a proxy score, classifier, or energy function indicating how strongly a

sample matches the forget criterion. Rather than impose a brittle threshold that converts soft membership into noisy labels, EBR reshapes the effective training distribution by assigning lower weight to high-energy, forget-associated samples, softly suppressing regions likely to contain forgotten data while preserving the data distribution elsewhere. It is therefore most appropriate when forget membership is uncertain, proxy-defined, or available only through an energy-based score.

5 Loss level Flow Decomposition

Building on the repulsion picture of Section 4, we induce the required repulsion *implicitly*, through a negative loss term, rather than by hand-crafting the velocity field. Our aim is to reproduce the model that retain-only training would yield without paying for that training, in closed form. The full-data risk splits *exactly* into retain and forget parts (Theorem 5.1), so the forget contribution cancels algebraically; rearranging isolates the retain-only risk (Corollary 5.2). That expression appears to need two model evaluations per step, but Theorem 5.3 collapses it to an unbiased *single-pass* estimator (so unlearning runs as a short fine-tune at one model call per update), and Theorem 5.4 lowers its variance at no cost in bias. In short, "retrain on the retained data" becomes a drop-in modification of the flow-matching loss.

5.1 Exact Retain-Risk Decomposition

Let $\mathcal{E}_\theta(x,t;z) = \|u_t^\theta(x) - u_t(x \mid z)\|_2^2$ be the conditional flow-matching error. For an endpoint distribution Q, set

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{\substack{z\sim Q,\ t\sim\mathrm{Unif}[0,1] \\ x\sim P_t(\cdot|z)}}[\mathcal{E}_\theta(x,t;z)]. \tag{1}$$

Instantiating Q at the full, retain, and forget endpoint laws defines the risks $\mathcal{L}_{\text{full}} = \mathcal{L}_{P_{\text{data}}}$, $\mathcal{L}_{\text{retain}} = \mathcal{L}_{P_{\text{retain}}}$, and $\mathcal{L}_{\text{forget}} = \mathcal{L}_{P_{\text{forget}}}$ used below. Write $\pi_R = \Pr(R = 1)$ for the proportion of retained data in the mixture, and $\pi_F = 1 - \pi_R$ for the forget proportion.

Theorem 5.1 (Mixture decomposition)**.** If $P_{\text{data}} = \pi_R P_{\text{retain}} + \pi_F P_{\text{forget}}$, then

$$\mathcal{L}_{\text{full}}(\theta) = \pi_R \mathcal{L}_{\text{retain}}(\theta) + \pi_F \mathcal{L}_{\text{forget}}(\theta).$$

The full-data risk is the retain/forget mixture of conditional risks, so the forget contribution can be removed algebraically rather than by retraining.

Corollary 5.2 (Exact retain-risk identity)**.** Under Theorem 5.1,

$$\mathcal{L}_{\text{retain}}(\theta) = \frac{\mathcal{L}_{\text{full}}(\theta) - \pi_F \mathcal{L}_{\text{forget}}(\theta)}{\pi_R}.$$

Computing the two terms separately takes two model evaluations. A defensive mixture instead combines them at one sampled bridge state.

Theorem 5.3 (One-pass defensive-mixture estimator)**.** Draw $z \sim P_{\text{data}}$, $z_f \sim P_{\text{forget}}$, and $x \sim M_t(\cdot \mid z, z_f)$, where $M_t = \lambda P_t(\cdot \mid z) + (1-\lambda) P_t(\cdot \mid z_f)$. Then, with importance weights $w_z = P_t(x \mid z)/M_t$ and $w_{z_f} = P_t(x \mid z_f)/M_t$ (writing $M_t = M_t(x \mid z, z_f)$), the exact retain risk equals

$$\mathcal{L}_{\text{retain}}(\theta) = \frac{1}{\pi_R}\,\mathbb{E}\big[w_z\,\mathcal{E}_\theta(x,t;z) - \pi_F\, w_{z_f}\,\mathcal{E}_\theta(x,t;z_f)\big]\,.$$

The estimator is unbiased, has the same expected gradient as exact retain-only training, and evaluates u_t^θ only once per sampled state.

The density ratios assign each sampled state back to its full-data and forget bridges, allowing their signed contributions to share one forward pass. The first endpoint z is drawn from the *full-data* law P_{data}, not P_{retain}; the retain risk is recovered by reweighting rather than by sampling retain points directly. The importance weights w_z and w_{z_f} use the conditional bridge densities $P_t(\cdot \mid z)$ in closed form, which are available for the Gaussian probability paths standard in flow matching.

Theorem 5.4 (Rao–Blackwellized estimator)**.** Let H be the one-branch estimator above and $C = (z, z_f, t, x_0)$ the sampled endpoints, time, and shared source. The estimator $H_{\text{RB}} = \mathbb{E}[H \mid C]$, obtained by evaluating and averaging both proposal branches, is unbiased and satisfies $\mathrm{Var}(H_{\text{RB}}) \le \mathrm{Var}(H)$.

Rao–Blackwellization removes proposal-branch randomness while preserving the target risk. LFD is derived natively for flow matching, where generation follows a deterministic velocity field: it removes the forget contribution directly from the flow-matching regression objective and evaluates the model only once per sampled state. This construction also specializes gracefully. When the generative process is instead a stochastic (SDE) diffusion and one adopts the score parameterization, the exact estimator ($\rho = 0$) coincides with SISS (Alberti et al., 2025), the corresponding diffusion-space method; Appendix E makes this correspondence precise. The two therefore agree in the stochastic-diffusion limit, while LFD additionally covers the deterministic-flow regime native to flow matching. Pseudocode and proofs are in Appendices A and B.3.

5.2 A practical acceleration

Exact LFD inherits the quality-preserving target of retain-only fine-tuning, but under a very short deletion budget it can approach that target slowly. As a simple remedy we optionally add a small, norm-bounded forget step. Let g_R be the gradient of the sampled LFD retain objective and g_F that of its importance-weighted forget component; we take

$$g_\rho = g_R - \rho \frac{\|g_R\|_2}{\|g_F\|_2 + \epsilon} g_F, \tag{2}$$

so the forget-directed step has the interpretable norm $\rho\|g_R\|_2$ regardless of the raw scale of g_F, keeping the preservation update dominant for $\rho < 1$. We apply this rule with either the one-branch Monte-Carlo estimator (*Weighted LFD-MC*) or its Rao–Blackwellized counterpart (*Weighted LFD-RB*), fixing $\rho = 0.10$ throughout; $\rho = 0$ recovers exact LFD. Exact LFD ($\rho = 0$) is the theoretical object of the previous section: the weighted rule ($\rho > 0$) deliberately adds a bounded forget-ascent term, and is therefore a modified objective rather than an unbiased estimator of the retain risk. We adopt it because exact LFD approaches the retain-only target slowly under the short deletion budgets (tens of updates) we operate at, and *all CelebA results we report use the weighted variant* (MC or RB). Because the forget term keeps descending the forget objective, a larger ρ or a longer budget can push forgetting *past* the exact-retraining reference (an aggressive-erasure regime rather than retain-equivalence), so ρ trades off erasure strength against fidelity to the retrain target.

6 Energy-Based Reweighting

Suppose explicit retain and forget examples are unavailable during unlearning, but an accessible endpoint law q_0 can be sampled and a frozen proxy estimates retain membership. Write $R = 1$ for "retain," let $r(z) = \Pr(R = 1 \mid Z = z)$ under q_0, and let $\widehat{r}(z) \in [0, 1]$ be the proxy estimate. EBR uses the forget-oriented energy

$$F_{\widehat{r}}(z) = -\log \widehat{r}(z) = -\log\big(1 - \widehat{p}_{\mathrm{F}}(z)\big), \tag{3}$$

with the convention $-\log 0 = +\infty$. Thus, forget-like endpoints have high energy and receive weight $e^{-\beta F_{\widehat{r}}(z)} = \widehat{r}(z)^\beta$, with no threshold on uncertain scores and no explicit forget samples in the update.

For $\beta > 0$, define $Z_\beta = \mathbb{E}_{q_0}[\widehat{r}(Z)^\beta]$ and assume $0 < Z_\beta < \infty$. EBR targets

$$q_{\mathrm{EBR},\beta}(z) = \frac{q_0(z)\widehat{r}(z)^\beta}{Z_\beta}. \tag{4}$$

The endpoint law q_0 may be an empirical data law when those samples are available; in our proxy-only experiments it is the frozen deployed-generator bank.

6.1 Theoretical basis

Theorem 6.1 (Variational characterization)**.** Among distributions q absolutely continuous with respect to q_0, the unique minimizer of

$$D_{\mathrm{KL}}(q\|q_0) + \beta\mathbb{E}_q[F_{\widehat{r}}(Z)] \tag{5}$$

is $q_{\mathrm{EBR},\beta}$ in Equation 4.

This is the Gibbs variational identity: the KL penalty preserves the accessible base law while the energy term suppresses endpoints that the proxy considers unlikely to be retained.

Theorem 6.2 (Weighted flow-matching objective)**.** The population conditional flow-matching risk for the endpoint law in Equation 4 is

$$\mathcal{L}_{\mathrm{EBR}}(\theta) = \frac{1}{Z_\beta} \mathbb{E}_{\substack{z\sim q_0,\ t\sim \mathrm{Unif}[0,1] \\ x\sim P_t(\cdot\mid z)}} \left[\widehat{r}(z)^\beta \left\| u_t^\theta(x) - u_t(x \mid z) \right\|_2^2 \right]. \tag{6}$$

Under the usual population flow-matching assumptions, its optimal velocity field generates $q_{\mathrm{EBR},\beta}$.

The unknown Z_β does not require separate estimation: the implementation uses the self-normalized minibatch loss $\sum_b w_b \ell_b / \sum_b w_b$ with $w_b = \widehat{r}(z_b)^\beta$. This ratio estimator is consistent but generally biased at finite batch size.

At $\beta = 1$ with a calibrated proxy ($\widehat{r} = r$) the reweighted law equals the exact retain law, while a proxy that discards retain–forget information incurs a strict, quantified density-ratio gap; explicit labels thus weakly, but not strictly, dominate proxy-only access, so the observed EBR shortfall from exact retraining need not stem from soft weighting alone. We state this oracle-equivalence boundary, with its calibration, prior-shift, finite-sample, and optimization caveats, as Proposition B.3 in Appendix B.5. Our MNIST run fixes $\beta = 1$ with a high-accuracy (99%) proxy; a systematic β and proxy-quality sweep is left to future work. Proofs and the prior correction used in the MNIST experiment appear in Appendix B.5; pseudocode appears in Appendix A.

7 Experiments

We evaluate each method in the access regime that motivates it. When the forget set is *explicitly available*, LFD gives a direct retain-only estimator; we use it for precise *identity* removal on CelebA. When no forget set is available and forgetting is specified only through a *proxy*, EBR reshapes the endpoint law from proxy scores alone; we use it to remove an entire *digit class* from MNIST. The two settings are complementary: identity deletion is a fine-grained, targeted request, whereas class deletion is a coarse, population-level one.

Protocol. Every comparison uses five independently trained unlearning seeds that start from identical base parameters and share Gaussian evaluation sources. We report two-sided 95% Student-t intervals over seeds and test directional consistency with an exact one-sided sign-flip test over the 2^5 seed sign assignments, so the smallest attainable p-value is $1/32 = 0.03125$. Exact retraining on the retained data alone (*gold*) is our behavioural reference: a target the unlearned model should approach, not a "lower-is-always-better" bound. Retained-set FID (the Fréchet Inception Distance between generated and real retained images, lower is better) measures how much of the original generative quality survives.

Methods and baselines. Three references recur throughout: the *base* model (trained on retain and forget data together), *naive deletion* (continued fine-tuning on the retained data alone, with no forget-subtraction term), and *gold* (exact retraining from scratch on the retained data, the target the unlearned model should approach). For LFD on CelebA we report the two Weighted-LFD estimators of Section 5.2: *Weighted LFD-MC* (*MC*), the one-branch Monte-Carlo estimator of the retain risk, and *Weighted LFD-RB* (*RB*), its Rao–Blackwellized counterpart that analytically averages both proposal branches to reduce variance at no added bias. For EBR on MNIST the external baseline is *ContinualFlow* (Simone et al., 2025), a post-hoc reweighting scheme whose energy-reweighting rule we reimplement on our endpoint bank (its public release is 2D-only).

7.1 LFD: precise identity removal on CelebA

Removing a named individual from a face generator is exactly the targeted, high-precision request that data-governance motivates, and the setting where an explicit forget set is naturally available, making it the natural home for LFD. We use six held-out CelebA identities of 30 images each (Liu et al., 2015) and study both the six single-identity requests and one joint request that deletes all 180 images at once. Forgetting is measured by FaceNet target similarity: the cosine similarity between

Table 1: **Identity removal preserves retained-set image quality.** Retained-split FID (lower is better) at the reported operating point (60 updates; naive deletion at its update-132 checkpoint); Δ is relative to the base model. Weighted LFD keeps generative quality intact (within 5% of base here and ≤ 15% at every intermediate checkpoint) while matching exact retraining (gold). Naive deletion keeps FID low only by barely perturbing the model; it fails to forget (Figure 2), so FID must be read alongside target similarity.

Method	Retained-set FID ↓	Δ vs. base
Base	27.8	0.0%
Gold (exact retrain)	27.7	−0.4%
Naive deletion	25.4	−8.5%
Weighted LFD-MC	**26.4**	−5.0%
Weighted LFD-RB	28.8	+3.7%

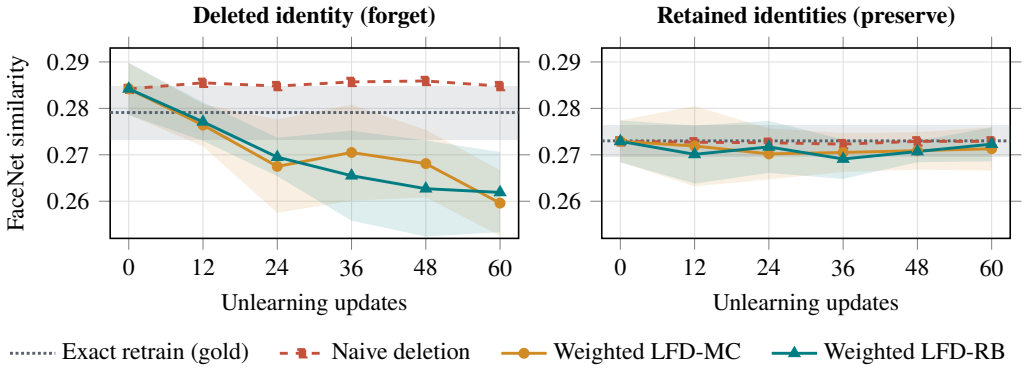

Figure 2: **LFD erases the requested face and leaves the rest intact.** *Left:* FaceNet similarity to the *deleted* identity falls to and below the exact-retraining reference (gold) under Weighted LFD, while naive deletion never moves it. *Right:* similarity to the *retained* identities stays at its base level throughout, so the erasure is targeted, not global. Both panels share the same axis; shaded regions are 95% Student-*t* intervals over five model seeds.

the generator's reconstruction of a held-out target image and that identity's embedding (Schroff et al., 2015); lower means the identity is harder to recover. We freeze $\rho = 0.10$, use five model seeds, and compare against naive retain-only deletion and against exact retraining.

Figure 2 reports the joint-six request. Both Weighted-LFD estimators reduce target similarity below naive deletion in *all five* seeds (mean advantage 0.026, 95% CI $[0.021, 0.031]$ for MC; 0.024, $[0.018, 0.030]$ for RB; each exact $p = 0.03125$), reaching the exact-retraining level. Yet the erasure barely disturbs other identities: retained-identity similarity shifts by only −0.002 (MC) and −0.001 (RB) versus naive (Figure 2, right). We take the joint-six deletion as the primary test and defer the six single-identity requests (same pattern) to Appendix D.2; retained-split FID also stays within a few percent of the base model (Table 1).

7.2 EBR: Removing a digit class on MNIST

Coarser requests ("forget an entire category") often arrive *without* an explicit forget set: we hold only a proxy that scores how forget-like a sample is. MNIST parity removal is an artificial but clean and controllable stand-in for attribute/class-level forgetting: we ask the generator to stop producing odd (resp. even) digits, supplying only a frozen parity proxy (a ten-class network reduced to parity, 99.0% accuracy, 0.9993 AUC) rather than any labelled forget images. A bank of 65,536 base-model endpoints is scored once, and each updated model sees only endpoints and frozen proxy weights. We run five seeds for 300 updates in each direction and evaluate 30,720 paired samples per system.

EBR removes the targeted parity strongly from proxy signal alone (Table 2): it cuts the generated odd-digit fraction from 49.3% to 10.7% and the even-digit fraction from 50.7% to 8.8%, moving most

Table 2: **EBR removes a whole digit-parity class from proxy scores alone, where a ContinualFlow baseline cannot.** Forgotten fraction is the generated probability of the requested parity (lower is stronger forgetting); feature FID is to the retained split (lower is better). ContinualFlow reweights the same endpoint bank yet barely moves the parity; EBR approaches exact retraining (gold). Five seeds.

	Forget odd		Forget even	
Method	Forgotten ↓	Feat. FID ↓	Forgotten ↓	Feat. FID ↓
Base	49.3%	357.5	50.7%	416.4
Gold (exact retrain)	1.0%	1.8	0.8%	4.3
ContinualFlow	48.1%	311.4	46.4%	301.7
EBR (ours)	**10.7%**	**51.7**	**8.8%**	**43.1**



Figure 3: **EBR removes even digits and leaves odd digits alone.** Paired samples from identical Gaussian seeds, before (Base) and after even-forget EBR. *Left:* seeds the base model renders as even digits are redrawn as sharp odd digits; the parity is forgotten. *Right:* seeds that render as odd digits are essentially unchanged; the retained parity is preserved. Table 2 quantifies both directions.

of the way toward exact retraining (≈ 1%) without ever seeing a labelled forget digit, while keeping retained-class quality intact (feature FID 43–52). The ContinualFlow baseline, which reweights the same endpoint bank, barely shifts the parity (only to 46–48%), so the removal comes from EBR's energy reweighting rather than the endpoint bank alone. Figure 3 shows paired samples before and after unlearning: the forgotten parity is redrawn as the retained parity while the retained digits stay essentially unchanged.

8 Conclusion

We presented an objective-level framework for data unlearning in flow matching models. Our formulation separates two practical information regimes: when the retain–forget split is explicit, and when forgetting is specified only through a soft proxy signal. For the former, we introduced LFD, which directly estimates the retain-only flow-matching objective. For the latter, we introduced EBR, which suppresses forget-associated samples through energy-based reweighting. Empirically, LFD provides strong weight-level unlearning when explicit splits are available, while EBR offers a lightweight alternative for proxy-defined forgetting. Together, these methods provide a principled and practical route to reducing forget-data influence while preserving retained generative behavior.

This paper establishes objective-level unlearning as a principled and workable route for flow matching, but it opens more questions than it settles. Data unlearning is a genuinely intricate problem, and charting its full landscape will take considerably more experimentation, from larger models and datasets to richer forget specifications, temperature and proxy-quality sweeps for EBR, and comparisons against stronger external baselines. We see this framework as a foundation for that broader study rather than a final word, and we hope it invites further exploration of how learned generative dynamics can be reshaped to forget.

AI use statement

Generative AI tools were used for LaTeX restructuring, language editing, and standard implementation assistance. They were not used to create experimental measurements. The authors reviewed the resulting text and code and checked all reported values against persisted experiment artifacts.

Ethics statement

Unlearning can support legitimate deletion requests, but empirical suppression does not itself provide a privacy guarantee. Misrepresenting approximate unlearning as certified removal could expose affected users to harm. We report the access assumptions, evaluator limitations, negative results, and exact retraining gaps needed to avoid that interpretation.

Reproducibility statement

Appendix D records dataset construction, optimization, evaluation, and independent replication units. All main comparisons use persisted checkpoints and paired noise, and the reported uncertainty is computed over model seeds rather than generated images. Code to reproduce all results will be released in a public GitHub repository upon acceptance.

References

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

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

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.

Xavier F Cadet, Anastasia Borovykh, Mohammad Malekzadeh, Sara Ahmadi-Abhari, and Hamed Haddadi. Deep unlearn: Benchmarking machine unlearning for image classification. In *2025 IEEE 10th European Symposium on Security and Privacy (EuroS&P)*, pp. 939–962. IEEE, 2025.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pp. 463–480. IEEE, 2015.

Court of Justice of the European Union. Judgment of the court (grand chamber) of 13 may 2014: Google spain sl and google inc. v agencia española de protección de datos (aepd) and mario costeja gonzález. Court of Justice of the European Union, Case C-131/12, 2014.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

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. *arXiv preprint arXiv:2310.12508*, 2023.

Zhaoxin Fan, Nanxiang Jiang, Daiheng Gao, Shiji Zhou, and Wenjun Wu. Eraseanything++: Enabling concept erasure in rectified flow transformers leveraging multi-object optimization. *arXiv preprint arXiv:2603.00978*, 2026.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2426–2436, 2023.

Daiheng Gao, Shilin Lu, Wenbo Zhou, Jiaming Chu, Jie Zhang, Mengxi Jia, Bang Zhang, Zhaoxin Fan, and Weiming Zhang. Eraseanything: Enabling concept erasure in rectified flow transformers. In *Forty-second International Conference on Machine Learning*, 2025.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *European Conference on Computer Vision*, pp. 383–398. Springer, 2020.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning (ICML)*, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Peter Holderrieth and Ezra Erives. Introduction to flow matching and diffusion models, 2025. URL `https://diffusion.csail.mit.edu/`.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

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

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, 2015.

Ziyao Liu, Huanyi Ye, Chen Chen, Yongsen Zheng, and Kwok-Yan Lam. Threats, attacks, and defenses in machine unlearning: A survey. *IEEE Open Journal of the Computer Society*, 6:413–425, 2025.

Protection Regulation. Regulation (eu) 2016/679 of the european parliament and of the council. *Regulation (eu)*, 679(2016):10–13, 2016.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

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*, 2025. arXiv:2506.18747.

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

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.

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

Lulu Xue, Shengshan Hu, Wei Lu, Yan Shen, Dongxu Li, Peijin Guo, Ziqi Zhou, Minghui Li, Yanjun Zhang, and Leo Zhang. Towards reliable forgetting: A survey on machine unlearning verification. *ACM Computing Surveys*, 58(12):1–35, 2026.

Gong Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1755–1764, 2024.

Algorithm 1 Exact LFD-MC via a Defensive Mixture

Require: Full-data sampler $z \sim P_{\text{data}}$, forget sampler $z_f \sim P_{\text{data}}(\cdot \mid R = 0)$
Require: Retain fraction $\pi_R \in (0, 1)$, mixture weight $\lambda \in (0, 1)$, steps S, learning rate η
Require: Bridge $P_t(x \mid z)$ and conditional target velocity $u_t(x \mid z)$
Initialize parameters θ
for $i = 1$ to S **do**
  Sample $z \sim P_{\text{data}}$, $z_f \sim P_{\text{data}}(\cdot \mid R = 0)$
  Sample $t \sim \text{Unif}[0, 1]$
  Sample $b \sim \text{Bernoulli}(\lambda)$ and then $x \sim P_t(\cdot \mid z)$ if b=1, else $x \sim P_t(\cdot \mid z_f)$
  $M_t(x \mid z, z_f) \leftarrow \lambda P_t(x \mid z) + (1 - \lambda) P_t(x \mid z_f)$
  $w_z(x, t) \leftarrow \dfrac{P_t(x \mid z)}{M_t(x \mid z, z_f)}, \quad w_{z_f}(x, t) \leftarrow \dfrac{P_t(x \mid z_f)}{M_t(x \mid z, z_f)}$
  $\mathcal{E}_\theta(x, t; z) \leftarrow \left\| u_t^\theta(x) - u_t(x \mid z) \right\|_2^2$
  $\mathcal{E}_\theta(x, t; z_f) \leftarrow \left\| u_t^\theta(x) - u_t(x \mid z_f) \right\|_2^2$
  $\widehat{\ell} \leftarrow \dfrac{1}{\pi_R} \Big(w_z(x, t)\, \mathcal{E}_\theta(x, t; z) - \pi_F\, w_{z_f}(x, t)\, \mathcal{E}_\theta(x, t; z_f) \Big)$
  $\theta \leftarrow \theta - \eta \nabla_\theta \widehat{\ell}$
end for
return u_t^θ

Algorithm 2 Stability-Controlled Weighted LFD-MC/RB

Require: Full-data sampler $z \sim P_{\text{data}}$, forget sampler $z_f \sim P_{\text{forget}}$
Require: Retain fraction π_R, proposal probability λ, perturbation ratio ρ, floor ϵ
Require: Estimator mode $m \in \{\text{MC}, \text{RB}\}$, bridge sampler, model u_θ, steps S, learning rate η
for $i = 1$ to S **do**
  Sample z, z_f, t and shared bridge source x_0; construct coupled states $x_z \sim P_t(\cdot \mid z)$ and $x_f \sim P_t(\cdot \mid z_f)$ from x_0
  if m = MC **then**
    Sample $b \sim \text{Bernoulli}(\lambda)$ and set $\mathcal{B} \leftarrow \{(x_z, 1)\}$ if $b = 1$, else $\mathcal{B} \leftarrow \{(x_f, 1)\}$
  else
    $\mathcal{B} \leftarrow \{(x_z, \lambda), (x_f, 1 - \lambda)\}$ ▷ Rao–Blackwellize the proposal branch
  end if
  $\widehat{L}_{\text{pres}} \leftarrow 0$, $\widehat{L}_{\text{F}} \leftarrow 0$
  for each $(x, a) \in \mathcal{B}$ **do**
    $M_t(x) \leftarrow \lambda P_t(x \mid z) + (1 - \lambda) P_t(x \mid z_f)$
    $w_z \leftarrow P_t(x \mid z)/M_t(x)$, $w_f \leftarrow P_t(x \mid z_f)/M_t(x)$
    $\widehat{L}_{\text{pres}}$ += $a\, w_z \mathcal{E}_\theta(x, t; z)/\pi_R$
    $\widehat{L}_{\text{F}}$ += $a\, \pi_F w_f \mathcal{E}_\theta(x, t; z_f)/\pi_R$
  end for
  $\widehat{L}_R \leftarrow \widehat{L}_{\text{pres}} - \widehat{L}_{\text{F}}$
  $g_R \leftarrow \nabla_\theta \widehat{L}_R$, $g_F \leftarrow \nabla_\theta \widehat{L}_{\text{F}}$
  $s \leftarrow \text{stopgrad}(\rho \|g_R\|_2 / (\|g_F\|_2 + \epsilon))$
  $\theta \leftarrow \theta - \eta\, (g_R - s g_F)$
end for
return u_θ

A Algorithm Pseudocode

B Proofs and estimator details

B.1 Assumptions

Assumption B.1 (Mixture and support)**.** The endpoint law is $P_{\text{data}} = \pi_R P_{\text{retain}} + \pi_F P_{\text{forget}}$ with $0 < \pi_R < 1$ and $\pi_F = 1 - \pi_R$. For almost every sampled (t, z_x, z_f), the bridge laws $P_t(\cdot \mid z_x)$ and

Algorithm 3 Energy-Based Reweighting for Flow-Matching Unlearning (EBR-Unlearn)

Require: accessible endpoint sampler $z \sim q_0$; frozen retain-probability proxy $\widehat{r}(z) = 1 - \widehat{p}_{\mathrm{F}}(z)$; exponent $\beta > 0$ and denominator floor $\epsilon > 0$; bridge family $P_t(x \mid z)$ and conditional target velocity $u_t(x \mid z)$; initial (deployed) velocity model $u_{\theta_0}(x, t)$; optimizer Opt; batch size B; unlearning steps T.

Ensure: Unlearned velocity model $u_{\theta_T}(x, t)$ approximating training on the energy-reweighted distribution $q_{\mathrm{EBR},\beta}(z) \propto q_0(z)\widehat{r}(z)^{\beta}$.

Initialize: $\theta \leftarrow \theta_0$. ▷ start from deployed model
for $k = 1$ to T **do**
  Sample minibatch $\{z_b\}_{b=1}^{B} \sim q_0$.
  Sample $\{t_b\}_{b=1}^{B} \sim \mathrm{Unif}[0, 1]$.
  for $b = 1$ to B **do**
    Sample $x_b \sim P_{t_b}(\cdot \mid z_b)$.
    Set $w_b \leftarrow \widehat{r}(z_b)^{\beta} = \exp\big(-\beta[-\log \widehat{r}(z_b)]\big)$.
    Compute target velocity $v_b \leftarrow u_{t_b}(x_b \mid z_b)$.
    Compute model prediction $\hat{v}_b \leftarrow u_\theta(x_b, t_b)$.
  end for
  Compute self-normalized reweighted flow-matching loss

$$\mathcal{L}_{\mathrm{EBR}}(\theta) \leftarrow \frac{\sum_{b=1}^{B} w_b \, \|\hat{v}_b - v_b\|_2^2}{\max\{\sum_{b=1}^{B} w_b, \epsilon\}}.$$

  Update $\theta \leftarrow \mathrm{Opt}(\theta, \nabla_\theta \mathcal{L}_{\mathrm{EBR}}(\theta))$.
end for
return u_{θ_T}

$P_t(\cdot \mid z_f)$ admit densities with respect to a common dominating measure. The defensive proposal has $\lambda \in (0, 1)$, so it is positive wherever either component is positive.

Assumption B.2 (Integrability and differentiation)**.** The weighted losses below are integrable, and differentiation with respect to θ may be interchanged with the endpoint, time, bridge, and proposal expectations.

These assumptions state the support and gradient-exchange conditions we make explicit here. Linear Gaussian bridges satisfy the common-support condition for every $t < 1$; implementations clip $1 - t$ away from zero.

B.2 Full-minus-forget identity

For any integrable endpoint functional $h_\theta(z) = \mathbb{E}_{t,x_t|z}[\ell_\theta(x_t, t; z)]$, the law of total expectation gives

$$\begin{aligned}
\mathcal{L}_{\mathrm{full}}(\theta) &= \mathbb{E}_{z\sim P_{\mathrm{data}}}[h_\theta(z)] && (7)\\
&= \pi_R \mathbb{E}_{z\sim P_{\mathrm{retain}}}[h_\theta(z)] + \pi_F \mathbb{E}_{z\sim P_{\mathrm{forget}}}[h_\theta(z)] && (8)\\
&= \pi_R \mathcal{L}_{\mathrm{retain}}(\theta) + \pi_F \mathcal{L}_{\mathrm{forget}}(\theta). && (9)
\end{aligned}$$

Rearranging proves Corollary 5.2. Under the differentiation assumption, the same identity holds for expected gradients.

B.3 Defensive-mixture estimator

Fix (z_x, z_f, t) and abbreviate $M = \lambda p_x + (1 - \lambda) p_f$. Importance sampling gives

$$\mathbb{E}_{y\sim M}\left[\frac{p_x(y)}{M(y)}\ell_\theta(y, t; z_x)\right] = \mathbb{E}_{y\sim p_x}[\ell_\theta(y, t; z_x)], \tag{10}$$

$$\mathbb{E}_{y\sim M}\left[\frac{p_f(y)}{M(y)}\ell_\theta(y, t; z_f)\right] = \mathbb{E}_{y\sim p_f}[\ell_\theta(y, t; z_f)]. \tag{11}$$

Taking expectations over $z_x \sim P_{\text{data}}$, $z_f \sim P_{\text{forget}}$, and t turns the two right-hand sides into $\mathcal{L}_{\text{full}}$ and $\mathcal{L}_{\text{forget}}$. Substitution into the estimator of Theorem 5.3 and the full-minus-forget identity prove unbiasedness. Differentiating the equality proves equality of expected gradients.

The first endpoint must be drawn from P_{data}, not P_{retain}. Drawing it from P_{retain} would estimate $(\mathcal{L}_{\text{retain}} - \pi_F \mathcal{L}_{\text{forget}})/\pi_R$, which is not the retain risk. This distinction is enforced in the production sampler.

B.4 Rao–Blackwellization

Let B be the proposal-branch indicator and let H denote the one-branch estimator in Theorem 5.3. Conditional on $C = (z_x, z_f, t, x_0)$,

$$\mathbb{E}[H \mid C] = \lambda H(y_x) + (1-\lambda) H(y_f), \tag{12}$$

which is the Rao–Blackwellized estimator of Theorem 5.4. The tower property preserves its expectation. The law of total variance gives

$$\operatorname{Var}(H) = \operatorname{Var}(\mathbb{E}[H \mid C]) + \mathbb{E}[\operatorname{Var}(H \mid C)] \geq \operatorname{Var}(\mathbb{E}[H \mid C]), \tag{13}$$

establishing that analytic branch averaging cannot increase variance.

B.5 EBR derivations and proxy-information limit

Variational identity. Let $F_{\widehat{r}} = -\log \widehat{r}$ and $Z_\beta = \mathbb{E}_{q_0}[e^{-\beta F_{\widehat{r}}}] = \mathbb{E}_{q_0}[\widehat{r}^{\beta}]$. For any $q \ll q_0$ for which the terms are finite,

$$D_{\mathrm{KL}}(q \| q_0) + \beta \mathbb{E}_q[F_{\widehat{r}}] = D_{\mathrm{KL}}\left(q \middle\| \frac{q_0 e^{-\beta F_{\widehat{r}}}}{Z_\beta}\right) - \log Z_\beta. \tag{14}$$

Nonnegativity of KL proves Theorem 6.1, including uniqueness up to q_0-null sets. Substituting this endpoint law into the conditional flow-matching expectation gives Equation 6 and Theorem 6.2.

Proposition B.3 (Oracle equivalence and proxy limitation)**.** Let $\pi_R = \Pr(R = 1) > 0$. If $\beta = 1$ and $\widehat{r} = r$, then

$$q_{\mathrm{EBR},1}(z) = q_0(z \mid R = 1). \tag{15}$$

Let $S = g(Z)$ be all information exposed to a proxy and set $s(S) = \Pr(R = 1 \mid S)$. The best S-measurable posterior induces a squared density-ratio gap

$$\int \left(\frac{dq_0(\cdot \mid R=1)}{dq_0} - \frac{dq_S}{dq_0}\right)^2 dq_0 = \frac{\mathbb{E}[\operatorname{Var}(r(Z) \mid S)]}{\pi_R^2}, \quad \frac{dq_S}{dq_0} = \frac{s(S)}{\pi_R}. \tag{16}$$

The gap vanishes exactly when the proxy statistic is sufficient for retain membership, equivalently when $I(R; Z \mid S) = 0$.

Oracle posterior. Let $q_0(z, R)$ be the joint accessible law, $R = 1$ denote retention, $r(z) = \Pr(R = 1 \mid Z = z)$, and $\pi_R = \Pr(R = 1)$. Bayes' rule gives

$$q_0(z \mid R = 1) = \frac{q_0(z) r(z)}{\pi_R}, \qquad \mathbb{E}_{q_0}[r(Z)] = \pi_R. \tag{17}$$

Consequently, setting $\widehat{r} = r$ and $\beta = 1$ in Equation 4 recovers the retain endpoint law exactly. This is a population identity, not a claim that an estimated classifier is exact or that finite optimization reaches the population minimizer.

Information lost by a proxy. Suppose the proxy observes only $S = g(Z)$ and define its Bayes posterior $s(S) = \Pr(R = 1 \mid S) = \mathbb{E}[r(Z) \mid S]$. The retain law and the best S-measurable reweighted law q_S have density ratios

$$\frac{dq_0(\cdot \mid R=1)}{dq_0} = \frac{r(Z)}{\pi_R}, \qquad \frac{dq_S}{dq_0} = \frac{s(S)}{\pi_R}. \tag{18}$$

The conditional-expectation projection theorem therefore yields

$$\int \left(\frac{r(z) - s(g(z))}{\pi_R} \right)^2 q_0(dz) = \frac{\mathbb{E}[(r(Z) - s(S))^2]}{\pi_R^2} \tag{19}$$

$$= \frac{\mathbb{E}[\mathrm{Var}(r(Z) \mid S)]}{\pi_R^2}, \tag{20}$$

which proves Equation 16. For any learned score $\widehat{r}(S)$, orthogonality further gives

$$\mathbb{E}[(r(Z) - \widehat{r}(S))^2] = \mathbb{E}[\mathrm{Var}(r(Z) \mid S)] + \mathbb{E}[(s(S) - \widehat{r}(S))^2]. \tag{21}$$

The first term is irreducible proxy insufficiency; the second is estimation or calibration error.

Because S is a function of Z,

$$I(R; Z \mid S) = \mathbb{E}\big[D_{\mathrm{KL}}\big(\mathrm{Bern}(r(Z)) \| \, \mathrm{Bern}(s(S))\big)\big]. \tag{22}$$

Hence $I(R; Z \mid S) = 0$ exactly when $r(Z) = s(S)$ almost surely, which is also exactly when the density-ratio gap vanishes. If $s(S) \in [\epsilon, 1 - \epsilon]$ almost surely, the bound $D_{\mathrm{KL}}(P\|Q) \le \chi^2(P\|Q)$ additionally gives

$$\frac{\mathbb{E}[\mathrm{Var}(r(Z) \mid S)]}{\pi_R^2} \ge \frac{\epsilon(1-\epsilon)}{\pi_R^2} I(R; Z \mid S). \tag{23}$$

Thus explicit labels weakly dominate proxy-only access in population, but the comparison need not be strict: an exact sufficient soft posterior ties the retain target and can reduce Monte Carlo variance relative to hard selection.

Prior correction in the experiments. If $s_F(z)$ is a forget posterior learned under a balanced binary sampling prior, converting its odds to deployment priors gives

$$\frac{\widetilde{s}_F(z)}{1 - \widetilde{s}_F(z)} = \frac{\pi_F}{\pi_R} \frac{s_F(z)}{1 - s_F(z)}, \qquad \widehat{r}(z) = 1 - \widetilde{s}_F(z). \tag{24}$$

The MNIST parity experiment applies this correction before weighting.

C Implementation details

Stable bridge weights. For a linear Gaussian bridge, the density ratio depends on

$$\log k_t(x, z) = -\frac{\|x - tz\|_2^2}{2(1-t)^2}. \tag{25}$$

The shared normalizer cancels. We compute proposal denominators with `logaddexp` and exponentiate only log-ratios, avoiding raw high-dimensional Gaussian densities.

Coupling and branch accounting. MC and RB pair full and forget endpoints with the same t and source noise. RB concatenates the two branch states in one batched model call, but counts both states in compute accounting. Source alignment is checked through $y_x - y_f = t(z_x - z_f)$; the maximum observed error in our confirmation runs was 4.62×10^{-7}.

Weighted gradient. The scale in Equation 2 is detached before assigning parameter gradients. Consequently, weighted LFD is best understood as a per-update gradient rule, not as a claim that a single globally fixed scalar objective exists. We report the realized perturbation ratio and gradient norms. Exact LFD uses $\rho = 0$; all weighted results use $\rho = 0.10$ selected on development evidence before their confirmation runs.

Nomenclature. "Exact LFD" denotes the unbiased retain-risk estimator. "Weighted LFD-MC" and "Weighted LFD-RB" denote the norm-perturbed one-branch and Rao–Blackwellized updates. "EBR" denotes proxy-weighted synthetic self-distillation and does not use explicit forget endpoints during the update. These names are not interchangeable.

Compute. All full image experiments ran on one NVIDIA A100 GPU, and persisted final checkpoints store model and optimizer state together with the row-sampling, bridge, proposal-branch, Torch, and CUDA random-number-generator states used by the pipeline.

D Experimental protocols

Table 3: Core experimental protocols. "Samples" is the final generated evaluation bank per model.

Study	Forget request	Seeds	Updates	Samples	Flow steps	Evaluator
CelebA LFD	Six identities, 180 images	5	60	10,000	50	FaceNet/SSCD
MNIST EBR	Odd or even digits	5 each	300	30,720	50	Parity/digit net

D.1 CelebA identity deletion

The approved base multiset has 171,306 rows. Each of six frozen identities has 30 images. Single requests remove one identity; the joint request removes all 180 images, leaving 171,126 rows. Weighted methods use learning rate 5×10^{-5}, 256 states per update, and checkpoints 12, 24, 36, 48, and 60. Naive deletion is evaluated through update 132. Exact matched and joint gold models train for 30,000 updates from the approved shared initialization.

For each identity and checkpoint, evaluation uses 30 target and 30 retain images, four corruption repetitions, and 25 reconstruction steps. FaceNet measures identity recovery and SSCD measures image-copy similarity. Latent NLL uses the Stable Diffusion VAE latent space and Hutchinson probes. Standard FID/KID uses 10,000 real and generated samples; joint standard quality is available for unlearning seed 17 only. Identity measurements are averaged within model seed before intervals and sign-flip tests.

D.2 Single-identity CelebA deletion

Beyond the joint request of Section 7.1, we also run the six identities as separate single-deletion requests (one 30-image identity each, five model seeds). Weighted LFD-MC and LFD-RB reduce target FaceNet similarity below naive deletion in all five seeds on every identity. At the update-12 checkpoint (Table 4), both estimators sit *below* each identity's matched-gold model: mean target FaceNet is 0.2595 (MC) and 0.2592 (RB) against a matched-gold mean of 0.2844, whereas naive deletion stays at 0.2910 and fails to forget. The estimators' signed gaps to matched gold near -0.025 mean they already overshoot exact retraining at the first measured checkpoint, while retaining image quality close to matched gold (FID +4.8% for MC and +10.2% for RB; naive -5.0%).

Unlike the joint request, which is near gold at the same update budget, a single 30-image request is a smaller forget mass, so a fixed update count erases more aggressively. The erasure budget (update count or ρ) should therefore scale with forget-set size rather than be held fixed across request sizes; we report the joint request as the primary result and treat these single-deletion runs as secondary, without multiple-comparison correction.

Table 4: Single-identity CelebA deletion: six separate 30-image requests over five model seeds (Weighted LFD at update 12; naive deletion at update 132). Target FaceNet similarity is lower-is-stronger forgetting; FID is relative to each identity's matched-gold retraining.

Method	Target FaceNet ↓	FID / matched gold
Matched gold (exact retrain)	0.2844	1.000
Naive deletion	0.2910	0.950
Weighted LFD-MC	0.2595	1.048
Weighted LFD-RB	0.2592	1.102

D.3 MNIST parity EBR

The foundation and odd-only/even-only gold models train for 117,500 updates. The proxy is independently initialized and trained with ten-class supervision; parity scores are log-sum-exp differences over odd and even logits. Its frozen validation accuracy and AUC are 0.9900 and 0.9993. The base model generates a 65,536-endpoint bank. EBR, synthetic distillation, and the two local ContinualFlow variants use the same bank, 300 updates, batch 64, learning rate 5×10^{-5}, and seeds 17, 31, 43, 59, and 73. Evaluation uses 30,720 paired sources. The reverse direction is a separate target condition within each seed, not an additional independent seed.

E LFD and SISS: deterministic flows versus stochastic diffusions

LFD is formulated for flow matching, where samples are generated by integrating a deterministic velocity field u_t along the probability-flow ODE $\dot{x} = u_t(x)$. SISS (Alberti et al., 2025) is formulated for diffusion models, where samples follow a stochastic differential equation driven by a score $\nabla \log P_t$. We show the two coincide exactly when the generative process is the stochastic one and the velocity is written in the score parameterization; LFD additionally covers the deterministic-flow case, which has no score-based SISS analog.

Shared retain-risk identity. Both methods rest on the same population fact. With $P_{\text{data}} = \pi_R P_{\text{retain}} + \pi_F P_{\text{forget}}$, any per-sample conditional objective $\mathcal{E}_\theta(x, t; z)$ that is linear in the endpoint law satisfies

$$\mathcal{L}_{\text{retain}}(\theta) = \frac{\mathcal{L}_{\text{full}}(\theta) - \pi_F \mathcal{L}_{\text{forget}}(\theta)}{\pi_R}. \tag{26}$$

LFD instantiates $\mathcal{E}_\theta$ with the flow-matching regression error $\|u_t^\theta(x) - u_t(x \mid z)\|_2^2$; SISS instantiates it with the denoising-score-matching error $\|s_\theta(x,t) - \nabla_x \log P_t(x \mid z)\|_2^2$.

The score parameterization links the two. For a Gaussian probability path $P_t(\cdot \mid z) = \mathcal{N}(\alpha_t z, \sigma_t^2 I)$ shared by both frameworks, the flow-matching velocity and the score are affine reparameterizations of one another,

$$u_t(x \mid z) = a_t\, x + b_t\, \nabla_x \log P_t(x \mid z), \tag{27}$$

for schedule-dependent scalars a_t, b_t. Substituting Equation 27 turns each velocity-matching term into a score-matching term up to the common factor b_t^2, so the LFD retain estimator of Equation 26 becomes exactly the (unweighted) SISS retain estimator in the score parameterization. The norm-bounded rule of Section 5.2 corresponds, under this mapping, to SISS's norm-controlled forget term.

Where they differ. The correspondence requires a stochastic (SDE) sampler whose marginal score is well defined. When generation is the deterministic probability-flow ODE (the native flow-matching setting), there is no diffusion score to match, yet the velocity-level identity of Equation 26 still holds. LFD therefore applies in both regimes and reduces to SISS precisely in the stochastic-diffusion special case.

F Additional qualitative samples

F.1 MNIST parity EBR samples

G Limitations

Scope. Both estimators do exactly what they are designed for: LFD performs precise, seed-consistent identity removal while preserving other identities, and EBR removes an entire class from proxy signal alone. These are behavioural, output-level results (not certified-privacy guarantees), and we are extending the range of tasks and datasets in ongoing work.

First, none of the methods provides certified data removal or differential privacy. Output suppression and reconstruction resistance are empirical properties of the tested evaluator and attack, not proofs that all information about a training example is absent from the weights.



Figure 4: Paired pixel-space samples for the forget-odd EBR diagnostic. Red frames denote evaluator-predicted forgotten digits.



Figure 5: Paired pixel-space samples for the forget-even EBR diagnostic. Columns share Gaussian sources across methods.

Second, weighted LFD deliberately changes the exact retain-risk gradient. Its strength parameter and stopping time define an erasure–fidelity trade-off; the CelebA study shows that a fixed rule does not transfer unchanged across deletion-set sizes. Exact-retraining equivalence requires predeclared margins and fresh confirmation data.

Third, forgetting metrics depend on external models. FaceNet, SSCD, and digit/parity classifiers can miss semantic or instance-level leakage. We reduce this risk through independent evaluators, multiple metrics, paired samples, and reconstruction probes, but no finite battery is exhaustive.

Fourth, EBR's oracle identity requires a retain posterior calibrated for the accessible endpoint law and $\beta = 1$. A sufficient calibrated soft posterior can equal the retain target, so softness alone does not imply inferior unlearning. In practice, proxy insufficiency, calibration and prior shift, finite-bank self-normalization (a ratio-estimator bias of order $1/B$ in the batch size B that shrinks as the bank grows), and incomplete optimization can all create a gap; a target nearly absent from the bank also cannot supply an effective weighted signal. The local ContinualFlow systems do not reproduce the published latent image implementation, so their ranking is not a claim about the published method.

Finally, the experiments cover three image domains and relatively small flow architectures. Scaling to text-conditioned models, repeated deletion requests, and adversaries with unrestricted white-box analyses remains open.