BEM: LEARNING BREGMAN DIVERGENCES FOR CONCEPT ERASURE IN FLOW MATCHING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Concept erasure removes harmful, private, or copyrighted concepts from text-to-image models while preserving their other abilities. Leading models such as FLUX and Stable Diffusion 3 are trained with flow matching, and erasure methods for them fine-tune the model toward a guided teacher's velocities under a Euclidean loss. Yet the squared distance is only one member of a larger family: Bregman divergences are exactly the losses under which flow matching recovers the correct velocity field, and each member shapes differently what is erased and what is preserved. We introduce BEM (Bregman erasure matching), which is, to our knowledge, the first to learn this divergence through bi-level optimization, so that the fine-tuned model forgets the concept on unseen prompts while its unrelated generations stay unchanged. In the experiments we target the suppression of nudity, violence, real identities, and copyrighted characters. Under the evaluation protocol of GEM, BEM removes more of the concept than GEM on most settings, cutting the unsafe rate on bloody gore from 53.9% to 14.4% and the FID on nudity from 8.64 to 6.64, while retention holds on some concepts and degrades on others.

Content warning: this paper contains generated images of graphic violence and censored nudity.

1 INTRODUCTION

Text-to-image models based on flow matching are deployed at scale, and with them comes the problem of removing what they should not produce. A released checkpoint may reproduce nudity and graphic violence, the likeness of real people, and the style of copyrighted characters, all of which the operator of a service may be required to remove (Schramowski et al., 2023; Huang et al., 2024). Retraining on a filtered corpus costs as much as the original training run. The concept is therefore removed after the fact, by fine-tuning the model against itself, and this is the setting we consider.

Erasure by velocity matching. Recent work shares one construction, built from two copies of the model. At each latent a frozen teacher supplies a velocity for a prompt naming the concept ($\boldsymbol{u}_c$) and a velocity for an anchor prompt describing what should appear instead ($\boldsymbol{u}_a$). A target velocity is formed by extrapolating away from the concept, $\bar{\boldsymbol{u}} = \boldsymbol{u}_a + s\,(\boldsymbol{u}_a - \boldsymbol{u}_c)$, and a student (in practice a low-rank adapter) is fit to it on partially denoised latents. The construction goes back to ESD by Gandikota et al. (2023) and carries over to the rectified flow transformers that the current models use (Gao et al., 2025a; Fan et al., 2026; Grebe et al., 2026). These methods differ in how $\bar{\boldsymbol{u}}$ is built, i.e., in the anchor, the noise levels at which the regression is applied, and the reach s of the extrapolation, where $s = 0$ regresses onto the anchor itself, as CA (Kumari et al., 2023) does.

The unexamined component. Where these methods regress, they use the squared distance as error, which is oblivious to the direction in the velocity space. A concept, however, occupies a direction, namely the one along which the teacher separates the concept prompt from the anchor. Charging the orthogonal directions alike is what makes an erasure procedure trade the concept against everything else the model can draw, and the literature reports the consequence as a degradation of image fidelity (Gandikota et al., 2023; Grebe et al., 2026). Note that not every loss is admissible here. A regression of this form is a flow matching regression, and the losses under which such a regression recovers its target are known to be exactly the Bregman divergences (Theorem 2.1 below). Thus the squared distance is only one choice within a family (the choice generated by the quadratic potential $\varphi(\boldsymbol{v}) = \|\boldsymbol{v}\|^2$), and no erasure method varies this choice.

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.

This work. In this paper we keep the unlearning procedure of Grebe et al. (2026) unchanged, i.e., its LoRA (Hu et al., 2022) adapter, its frozen teacher, its window of noise levels and its optimizer, except in two respects: we replace its hinge loss by a regression under a Bregman divergence, which is generated by a strictly convex potential φ, and we add meta steps that learn φ. Intuitively, this means that the unlearning procedure is told what a velocity error costs along the direction that carries the concept, separately from what it costs along the directions that carry everything else, instead of being told that errors in all directions cost the same. The potential is learned through bi-level optimization: the inner problem erases the concept under the Bregman divergence generated by the current potential, and the outer problem minimizes a scoring function of the resulting model, so that the potential is chosen by how well its divergence erases the concept and retains everything else rather than by how well the erasure fits.

Contributions. This paper takes a known characterization of the valid losses for flow matching, observes that all erasure methods that regress make a single choice within it, and learns this choice instead of fixing it:

- We observe that erasure by velocity matching inherits the valid family of flow matching (Theorem 2.1), so that the divergence is a free parameter of every such method. Prior work sets it to the squared distance and does not vary it.
- We propose BEM, which parameterizes the potential as a bounded, context-dependent metric with a separate curvature along the direction that separates the concept from the anchor, and learns it by bi-level optimization with an implicit-function-theorem hypergradient.
- On six concepts and two model families we compare against the protocol with its other components held fixed, so that a difference is attributable to the loss, not to the many other choices that an erasure procedure involves.
- BEM erases more of the concept than GEM (Grebe et al., 2026), whose loss it replaces, on every concept we evaluate. On nudity it is better on five of the six benchmarks, on bloody gore it leaves 14.4% unsafe against 53.9%, at a lower FID in both, on the four rights-protected concepts it leaves at most 1% against up to 10%, and on Stable Diffusion 3.5 it leaves 3% against 17%. Retention rises on two of these concepts, holds on one, and falls on two.

2 Background

The models this paper edits, FLUX (Black Forest Labs et al., 2025) and Stable Diffusion 3 (Esser et al., 2024), are trained by flow matching. This section recalls its objective, introduces Bregman divergences, and states the result that ties the two together.

2.1 Flow Matching

A flow matching model generates an image by integrating an ordinary differential equation from a Gaussian sample $\boldsymbol{x}_0$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\,\boldsymbol{x}_t \;=\; \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t, c), \qquad t : 0 \to 1, \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^d$ is the latent at time t, c is the text prompt, $\boldsymbol{v}_{\boldsymbol{\theta}}$ is a neural network with parameters $\boldsymbol{\theta}$, and the image is $\boldsymbol{x}_1$. FLUX and Stable Diffusion 3 parameterize time so that $t = 0$ is data and $t = 1$ is noise; this flips the sign of the velocity and changes nothing below. Flow matching fixes the interpolation between noise and data by hand (Lipman et al., 2022; Liu et al., 2022): one draws a pair $\boldsymbol{z}$ of noise and data from a coupling and interpolates between the two, along the straight line for the rectified flows of the models above. The interpolant determines a conditional velocity $\boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{z})$ in closed form, but what generates the data is its average over all pairs that pass through the same latent,

$$\boldsymbol{u}_t(\boldsymbol{x}) \;=\; \mathbb{E}\big[\, \boldsymbol{u}_t(\boldsymbol{x} \mid \boldsymbol{z}) \;\big|\; \boldsymbol{x}_t = \boldsymbol{x} \,\big], \tag{2}$$

the marginal field. Throughout we reserve $\boldsymbol{u}$ for such reference velocities and $\boldsymbol{v}$ for the velocity of the model being trained. Regressing the model onto the marginal field ($\mathcal{L}_{\mathrm{FM}}$ below) is intractable, since the average in Eq. (2) cannot be evaluated, and replacing it by the conditional field gives the

tractable $\mathcal{L}_{\mathrm{CFM}}$,

$$\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) = \mathbb{E}\big[\|\boldsymbol{u}_t(\boldsymbol{x}_t) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|^2\big], \qquad \mathcal{L}_{\mathrm{CFM}}(\boldsymbol{\theta}) = \mathbb{E}\big[\|\boldsymbol{u}_t(\boldsymbol{x}_t \mid \boldsymbol{z}) - \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|^2\big], \tag{3}$$

with expectations over t, $\boldsymbol{x}_t$ and, in the second, $\boldsymbol{z}$. Training is possible because the two have the same gradient (Lipman et al., 2022; 2024),

$$\nabla_{\boldsymbol{\theta}}\, \mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) \;=\; \nabla_{\boldsymbol{\theta}}\, \mathcal{L}_{\mathrm{CFM}}(\boldsymbol{\theta}), \tag{4}$$

and nothing so far requires the squared distance itself.

2.2 Bregman Divergences

Let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be strictly convex and continuously differentiable. The Bregman divergence generated by φ (Bregman, 1967) is the gap at $\boldsymbol{u}$ between φ and its linearization at $\boldsymbol{v}$,

$$D_\varphi(\boldsymbol{u}, \boldsymbol{v}) \;=\; \varphi(\boldsymbol{u}) - \varphi(\boldsymbol{v}) - \langle \nabla\varphi(\boldsymbol{v}),\, \boldsymbol{u} - \boldsymbol{v}\rangle. \tag{5}$$

It is nonnegative, vanishes exactly at $\boldsymbol{u} = \boldsymbol{v}$, and is in general not symmetric. Two choices of φ are familiar: $\varphi(\boldsymbol{v}) = \|\boldsymbol{v}\|^2$ gives the squared distance $\|\boldsymbol{u} - \boldsymbol{v}\|^2$, and the negative entropy gives the Kullback-Leibler divergence. Intuitively, φ fixes how the cost of an error grows with its direction and magnitude, and the squared distance is the choice that makes the cost the same in every direction.

2.3 The Valid Losses for Flow Matching

The two meet in a known result. Replace the squared distance in Eq. (3) by a general loss D,

$$\mathcal{L}^D_{\mathrm{FM}}(\boldsymbol{\theta}) = \mathbb{E}\big[\, D\big(\boldsymbol{u}_t(\boldsymbol{x}_t),\, \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\big)\big], \qquad \mathcal{L}^D_{\mathrm{CFM}}(\boldsymbol{\theta}) = \mathbb{E}\big[\, D\big(\boldsymbol{u}_t(\boldsymbol{x}_t \mid \boldsymbol{z}),\, \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\big)\big]. \tag{6}$$

The question is for which D the identity in Eq. (4) survives, since without it the tractable objective is no longer a proxy for the intractable one.

Theorem 2.1 (Bregman divergences are the valid losses)**.** *Let $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be smooth with $D(\boldsymbol{u}, \boldsymbol{v}) = 0$ if and only if $\boldsymbol{u} = \boldsymbol{v}$. Then*

$$\nabla_{\boldsymbol{\theta}}\, \mathcal{L}^D_{\mathrm{FM}}(\boldsymbol{\theta}) \;=\; \nabla_{\boldsymbol{\theta}}\, \mathcal{L}^D_{\mathrm{CFM}}(\boldsymbol{\theta}) \quad \textit{for all } \boldsymbol{\theta} \qquad \Longleftrightarrow \qquad D = D_\varphi \textit{ for a strictly convex } \varphi.$$

The forward direction is the standard conditional flow matching argument and is proved by Lipman et al. (2024), Section 4.5. The converse is the part that matters here, since it says that no loss outside the family Eq. (5) admits tractable training in this sense; it is proved by Holderrieth et al. (2024), Appendix C.3, from the characterization by Banerjee et al. (2005) of the losses whose population minimizer is the conditional mean. Thus the choice made by every erasure method, the squared distance, is a single choice within a family that is fixed by nothing but convention, and moving inside that family leaves training as tractable as it was.

3 Related Work

We review erasure in diffusion models, attacks on erased models and defenses against them, erasure in flow models, and the unlearning procedure of GEM on which we build. Tab. 10 in App. D lists these methods with the models they are evaluated on and what they minimize, and Kim & Qi (2025) survey the field more broadly.

Erasure in diffusion models. Methods for diffusion models edit the weights in closed form, fine-tune them or act at sampling time. Closed-form methods edit the cross-attention or the text encoder (Orgad et al., 2023; Arad et al., 2024; Gandikota et al., 2024; Gong et al., 2024; Lu et al., 2024). Among fine-tuning methods, ESD (Gandikota et al., 2023) introduced the teacher-guided regression of Sec. 1, with the empty prompt as anchor. CA (Kumari et al., 2023) and others (Kim et al., 2023; Lyu et al., 2024; Li et al., 2025; Bui et al., 2024) change its target or what it preserves. Further methods alter the training signal or the fine-tuned component (Ni et al., 2023; Zhang et al., 2024a; Poppi et al., 2024; Liu et al., 2024), or adapt machine unlearning (Heng & Soh, 2023; Fan et al., 2024; Wu & Harandi, 2024). DUO (Park et al., 2024) optimizes preferences, EraseFlow (Kusumba et al., 2025) a trajectory-balance criterion, and EraseDiff (Wu et al., 2025) a bi-level

problem. Note that where the methods above regress, they use the squared distance. Sampling-time methods leave the weights unchanged. They rewrite the prompt (Ni et al., 2024), change the guidance (Schramowski et al., 2023; Jain et al., 2024), project the text embeddings off an unsafe subspace (Yoon et al., 2025) or add learned concept directions to the activations (Li et al., 2024).

Robustness. Several attacks recover erased concepts. Ring-A-Bell (Tsai et al., 2024), P4D (Chin et al., 2024) and UnlearnDiff (Zhang et al., 2024c) optimize the prompt. Pham et al. (2024) learn token embeddings instead, and ReFlux (Jiang et al., 2025) reactivates erased concepts in FLUX. MMA-Diffusion (Yang et al., 2024) bypasses prompt filters and safety checkers. Defenses train against such attacks (Huang et al., 2024; Kim et al., 2024; Srivatsan et al., 2025; Zhang et al., 2024b) or resist fine-tuning back toward the concept (Zheng & Yeh, 2024; Gao et al., 2025b). We evaluate on the prompts of Ring-A-Bell, MMA-Diffusion and P4D but do not train against attacks.

Flow models. EraseAnything (Gao et al., 2025a) carries the teacher-guided construction to rectified flow transformers. It places erasure and preservation in the two levels of a bi-level problem, which EraseAnything++ (Fan et al., 2026) recasts as a constrained multi-objective problem. Other methods handle single-stream transformers (Jiang et al., 2026), regress the final outputs of the sampler through all steps (Zhang et al., 2025b), optimize erasure and preservation rewards (Sun et al., 2026a) or unlearn by flow matching toward a reweighted target (Simone et al., 2025). Without training, DVE (Zhang et al., 2026) projects the velocity off the direction in which the concept and anchor velocities differ, and Mosaic (Ko et al., 2026) blends the velocity fields of erased models per region.

The unlearning procedure we build on. GEM (Grebe et al., 2026) bridges the trajectory view of EraseFlow and the teacher-guided view of ESD. From the trajectory view it takes one observation: erasure should be reinforced across a segment of the generation path, not at one noise level. It therefore fixes a window, the first $t_{\text{stop}}+1$ steps of the sampler counted from the noise, and evaluates all the velocities in it in parallel. From the teacher-guided view it takes the frozen teacher, which supplies a safe and an unsafe velocity. Its objective, however, is not a regression. Write d_{positive} for the distance of the student to the teacher's velocity on the anchor prompt, and d_{negative} for its distance to the teacher's velocity on the concept prompt. GEM minimizes the contrastive hinge

$$\mathcal{L}_{\text{GEM}}(\boldsymbol{\theta}) \;=\; \max\big(0,\; d_{\text{positive}} - \eta\, d_{\text{negative}}\big), \tag{7}$$

which pulls the student toward the safe velocity and pushes it away from the unsafe one. A noise level switches off once $d_{\text{positive}} \le \eta\, d_{\text{negative}}$.

In contrast to these methods, BEM learns the loss. EraseAnything and EraseDiff optimize the model at both levels of their bi-level problems, DVE removes the concept direction at sampling time and GEM minimizes a hinge. The outer level of our bi-level problem instead learns the potential that defines the loss, which weights the concept direction inside a regression.

We build on GEM for three reasons. First, it is a recent state-of-the-art method: among the methods it compares with, it reports the lowest unsafe rates on all nudity and gore benchmarks we use (App. A.2). Second, its objective is a single loss on the velocities, so the loss is what defines the method, whereas other methods for flow models add further terms, such as attention terms or constraints. GEM is therefore the natural setting in which to study the loss itself. Third, its released code reproduces the reported results closely on most benchmarks (App. A.2), which we did not find for all the methods we considered. We therefore keep its unlearning procedure and swap the hinge loss of Eq. (7) for a learned Bregman divergence.

4 Learning the Divergence

We adopt the unlearning procedure of GEM (Grebe et al., 2026), i.e., its low-rank adapter on the attention projections of the frozen base model, its frozen teacher, its window of noise levels, its partially denoised latents and its optimizer, and replace the hinge loss of Eq. (7) by a regression under a Bregman divergence. From here on $\boldsymbol{\theta}$ denotes the parameters of the adapter and $\boldsymbol{v}_{\boldsymbol{\theta}}$ the velocity of the adapted model, the student. The teacher is queried twice at the same latent: we write $\boldsymbol{u}_c$ for its velocity on a prompt that names the concept, $\boldsymbol{u}_a$ for its velocity on an anchor prompt that describes what should appear instead, and $\Delta = \boldsymbol{u}_a - \boldsymbol{u}_c$ for the separation of the two. For a strictly convex potential φ, unlearning minimizes

$$\mathcal{L}_{\text{BEM}}(\boldsymbol{\theta}; \varphi) \;=\; D_{\varphi}\big(\bar{\boldsymbol{u}},\, \boldsymbol{v}_{\boldsymbol{\theta}}\big), \qquad \bar{\boldsymbol{u}} \;=\; \boldsymbol{u}_a + s\,\Delta, \tag{8}$$

where D_φ is the Bregman divergence of Eq. (5) and the student is queried on the concept prompt at the same latent as the teacher. The target $\bar{\boldsymbol{u}}$ is the anchor velocity extrapolated away from the concept by the reach $s \geq 0$. By Theorem 2.1, every D_φ is a valid flow matching loss, and $\varphi(\boldsymbol{v}) = \|\boldsymbol{v}\|^2$ gives the squared distance of the regressions in Sec. 3. Thus choosing the loss amounts to choosing φ. In the rest of this section we state how a potential is judged (Sec. 4.1), construct a family of potentials (Sec. 4.2), and select a potential from it by bi-level optimization (Sec. 4.3).

4.1 Judging a Potential

We now ask how to learn φ. Note that Eq. (8) cannot judge its own potential: zero loss means that the student matches $\bar{\boldsymbol{u}}$ where it was trained, and says nothing about how well it erases the concept or retains everything else. We therefore judge a potential by the model that results from unlearning with it. A good potential yields a student that erases the concept and retains everything else, and we measure both with the meta loss

$$\mathcal{L}_{\text{meta}}(\boldsymbol{\theta}) \;=\; \mathcal{L}_{\text{erasure}}(\boldsymbol{\theta}) \;+\; \rho\,\mathcal{L}_{\text{retention}}(\boldsymbol{\theta}), \tag{9}$$

evaluated at the adapter after unlearning. The weight ρ balances the erasure criterion $\mathcal{L}_{\text{erasure}}$ and the retention criterion $\mathcal{L}_{\text{retention}}$; we hold $\rho = 0.05$ on nudity and set it by the rule of App. C.2 elsewhere. Theorem 2.1 does not constrain Eq. (9): the theorem governs a loss that trains a velocity field, and Eq. (9) only scores one that is already trained. We detail the two criteria next.

Erasure criterion. Let $\boldsymbol{v}$ be the velocity of the student on the concept prompt (computed at its latent) and Δ the separation of the concept prompt from its anchor at that latent. We want to know how far the student has traveled from the concept toward the anchor, so we project its displacement onto Δ,

$$\kappa_i \;=\; \frac{\langle \boldsymbol{v}_i - \boldsymbol{u}_{c,i},\, \Delta_i \rangle}{\|\Delta_i\|^2}, \tag{10}$$

where i indexes the tokens of the latent. The projection is 0 at an unchanged token and 1 at one that has reached the anchor. Dividing by $\|\Delta_i\|^2$ makes κ_i scale free, and κ_i ignores motion orthogonal to Δ_i. The erasure criterion is

$$\mathcal{L}_{\text{erasure}}(\boldsymbol{\theta}) \;=\; \textstyle\sum_i m_i\, \sigma\big(1-\kappa_i\big), \qquad m_i \;=\; \|\Delta_i\|^2 \Big/ \textstyle\sum_j \|\Delta_j\|^2, \tag{11}$$

averaged over a batch of fresh latents, with $\sigma(x) = \text{softplus}(\beta x)/\beta$ a smoothed hinge (we use $\beta = 4$). The saliency weights m_i concentrate the criterion on the tokens that carry the concept; where $\|\Delta_i\|$ is small, the teacher barely distinguishes the two prompts and there is nothing to erase. The hinge saturates beyond $\kappa_i = 1$, so that a token need do little more than reach the anchor.

Retention criterion. The anchor prompt does not name the concept, so on it we want to preserve the teacher's velocity $\boldsymbol{u}_a$, and any deviation of the student's velocity $\boldsymbol{v}$ on the anchor prompt from $\boldsymbol{u}_a$ is damage. Here no direction is privileged, and the plain squared distance is the right measurement. The retention criterion is

$$\mathcal{L}_{\text{retention}}(\boldsymbol{\theta}) \;=\; \frac{1}{\bar{U}}\,\|\boldsymbol{v} - \boldsymbol{u}_a\|^2, \tag{12}$$

where $\bar{U}$ is the mean of $\|\Delta\|^2$ over the erasure latents, so that both criteria are dimensionless.

4.2 A Family of Potentials

Concept direction. We now ask which potentials can score well on Eq. (9). The squared distance penalizes an error of a given length the same in every direction. The meta loss, however, singles out one direction: Eq. (10) scores motion along Δ only, and retention charges every direction alike. Motion along Δ is erasure, since the target $\bar{\boldsymbol{u}}$ lies on that line. Motion orthogonal to Δ changes the velocity without removing the concept; it does not help erasure and, to the extent that it transfers to nearby prompts, harms retention. When the adapter cannot fit the target exactly, as an adapter of low rank trained for a few steps cannot, a potential that penalizes the two directions differently can therefore increase erasure at a fixed amount of damage. This is the freedom we exploit. We construct a family of potentials $\varphi_{\boldsymbol{\psi}}$, indexed by the parameters $\boldsymbol{\psi}$ of a neural network, with three properties:

(i) every $\boldsymbol{\psi}$ yields a strictly convex potential;

(ii) the family contains the squared distance;
(iii) the Hessian of every member is cheap to compute.

Property (iii) matters because the hypergradient in Sec. 4.3 applies the Hessian once per conjugate gradient iteration. A potential acts on the velocities of all tokens of a latent at one noise level.

Architecture. We follow the partially input convex networks of Amos et al. (2017) and let the tokens interact by attention. A context path reads the latent, the noise level, the two prompts and the separation Δ, and emits an encoding $\boldsymbol{h}_i$ of every token, from which it computes gates $\boldsymbol{\gamma}_k$, biases $\boldsymbol{b}_k$, attention weights $\boldsymbol{A}_k$, pooling weights $\pi_i \geq 0$ and two curvatures p_i and q_i per token. It never receives $\boldsymbol{v}$, so convexity in $\boldsymbol{v}$ places no constraint on it. The $\boldsymbol{v}$-path is a residual stack of K layers,

$$\begin{aligned} \boldsymbol{z}_1 &= g\Big(\boldsymbol{W}_0\big(\boldsymbol{v} \odot \boldsymbol{\gamma}_0\big) + \boldsymbol{b}_0 + \big(\boldsymbol{W}_0'\big(\boldsymbol{v} \odot \boldsymbol{\gamma}_0'\big) + \boldsymbol{b}_0'\big)^2\Big), \\ \boldsymbol{z}_{k+1} &= \boldsymbol{z}_k + g\Big(\boldsymbol{W}_k^{(z)}\boldsymbol{z}_k + \boldsymbol{W}_k^{(m)}\boldsymbol{A}_k\boldsymbol{z}_k + \boldsymbol{W}_k^{(v)}\big(\boldsymbol{v} \odot \boldsymbol{\gamma}_k\big) + \boldsymbol{W}_k^{(a)}\boldsymbol{A}_k\big(\boldsymbol{v} \odot \boldsymbol{\gamma}_k'\big) + \boldsymbol{b}_k\Big), \end{aligned} \tag{13}$$

for $k = 1, \ldots, K-1$, where $\boldsymbol{z}_k$ collects the hidden states of all tokens, $\boldsymbol{A}_k$ mixes them across tokens, the square is taken entrywise, g is a convex nondecreasing activation applied entrywise (we use softplus), and for readability we omit the context gates on $\boldsymbol{W}_k^{(z)}\boldsymbol{z}_k$ and $\boldsymbol{W}_k^{(m)}\boldsymbol{A}_k\boldsymbol{z}_k$. The potential of the latent is

$$\varphi_{\boldsymbol{\psi}}(\boldsymbol{v}) \;=\; \sum_i \pi_i\, \boldsymbol{w}^\top \boldsymbol{z}_{K,i} \;+\; \tfrac{1}{2}\sum_i \Big(q_i\|\boldsymbol{v}_i\|^2 + (p_i - q_i)\,\langle \boldsymbol{e}_i, \boldsymbol{v}_i\rangle^2\Big), \tag{14}$$

up to two further convex terms of the network, a log-sum-exp over context keys and a small convex network on learned poolings of $\boldsymbol{v}$. In words, the potential is a quadratic in $\boldsymbol{v}$ plus a convex network of $\boldsymbol{v}$, and the concept direction enters through the quadratic, via $\boldsymbol{e}_i = \Delta_i/\sqrt{\|\Delta_i\|^2 + \epsilon^2}$, so that $\|\boldsymbol{e}_i\| \leq 1$, with ϵ a small fraction of the mean of $\|\Delta_i\|$ over the tokens. The quadratic at token i has curvature q_i orthogonal to $\boldsymbol{e}_i$ and $q_i + (p_i - q_i)\|\boldsymbol{e}_i\|^2$ along it. Thus the curvature along $\boldsymbol{e}_i$ moves from q_i toward p_i as $\|\Delta_i\|$ grows. Unlike the quadratic, the network makes $\nabla^2\varphi_{\boldsymbol{\psi}}$ depend on $\boldsymbol{v}$.

Properties. The attention weights $\boldsymbol{A}_k$ and the pooling weights π_i are nonnegative and depend on the context alone, $\boldsymbol{W}_k^{(z)}$, $\boldsymbol{W}_k^{(m)}$ and $\boldsymbol{w}$ are nonnegative entrywise, and g is convex and nondecreasing, so the network is convex in $\boldsymbol{v}$ for every $\boldsymbol{\psi}$: every layer is a nonnegative combination of convex functions of $\boldsymbol{v}$ under a convex nondecreasing map. A fixed floor $\mu > 0$ with $p_i, q_i \geq \mu$ makes the quadratic strongly convex, and with it $\varphi_{\boldsymbol{\psi}}$. Thus every $\boldsymbol{\psi}$ yields a valid loss, which is property (i); the sign constraints are built into the parameterization, so nothing has to be enforced during training. Setting $p_i = q_i = 1$ and removing the network reduces $D_{\varphi_{\boldsymbol{\psi}}}$ to half the squared distance, so the family contains the loss of every regression in Sec. 3, which is property (ii). We initialize there, up to a calibration in which the network contributes $2 \cdot 10^{-3}$ of the curvature of the quadratic. Thus the search starts at the squared distance, and every subsequent difference is learned. Finally, the Hessian is cheap, which is property (iii): at token i the quadratic contributes $q_i\boldsymbol{I} + (p_i - q_i)\,\boldsymbol{e}_i\boldsymbol{e}_i^\top$ in closed form, and a Hessian-vector product of the network costs a few passes of automatic differentiation through it, which is small next to a pass through the transformer.

4.3 Selecting a Potential

Bi-level problem. Unlearning minimizes Eq. (8) and judging evaluates Eq. (9), so selecting a potential is the bi-level problem

$$\min_{\boldsymbol{\psi}} \; \mathcal{L}_{\mathrm{meta}}\big(\boldsymbol{\theta}^*(\boldsymbol{\psi})\big) \quad \text{subject to} \quad \boldsymbol{\theta}^*(\boldsymbol{\psi}) \;\in\; \arg\min_{\boldsymbol{\theta}} \; \mathcal{L}_{\mathrm{BEM}}\big(\boldsymbol{\theta};\, \varphi_{\boldsymbol{\psi}}\big), \tag{15}$$

over the family of Sec. 4.2. The outer variable $\boldsymbol{\psi}$ enters the inner problem only through the loss. We call the resulting method BEM (Bregman erasure matching). Differentiating Eq. (15) through the inner solution is the standard difficulty, and we use the implicit function theorem in the form of Rajeswaran et al. (2019). At an inner stationary point $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{BEM}} = 0$, and differentiating that identity in $\boldsymbol{\psi}$ gives

$$\nabla_{\boldsymbol{\psi}}\mathcal{L}_{\mathrm{meta}} \;=\; -\big(\nabla^2_{\boldsymbol{\theta}\boldsymbol{\psi}}\mathcal{L}_{\mathrm{BEM}}\big)^\top\big(\nabla^2_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{BEM}} + \lambda\boldsymbol{I}\big)^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathrm{meta}}, \tag{16}$$

with λ damping the inverse. Neither matrix is formed. We apply the inverse with conjugate gradient iterations and replace the Hessian by the Gauss-Newton form built from $\nabla^2\varphi_{\boldsymbol{\psi}}$, which is positive

Table 1: Safety after erasing ✗ nudity (left) and ✗ bloody gore (right) on FLUX. Lower is better in every column, and the better of the two erased models is in bold. The gore rates of BEM are a mean over three seeds, every other entry a single run. † marks a rate of BEM that differs from GEM's with $p < 0.05$ (two-sided Fisher exact test; App. A.3 gives 95% intervals). RAB is Ring-A-Bell.

	✗ nudity								✗ bloody gore			
Method	I2P	T2I-RP	RAB	MMA	P4D	Basic	FID	min	T2I-RP	Basic	FID	min
FLUX	18.05	52.04	66.67	27.70	44.49	76	0.00	0	84.27	88	0.00	0
GEM	6.87	**22.11**	27.72	4.60	15.07	8	8.64	4:05	53.85	**0**	4.91	7:56
BEM (Ours)	**5.59**	26.02†	**10.53**†	**3.00**	**4.78**†	**1**†	**6.64**	211:15	**14.36**†	0.33	**4.26**	107:10

semidefinite because φ_{ψ} is convex. Note that the exact Hessian of $D_{\varphi_{\psi}}$ in $\boldsymbol{v}_{\boldsymbol{\theta}}$ involves a third derivative of φ_{ψ} and is not positive semidefinite in general, which is why we do not use it.

Implementation and cost. The hypergradient in Eq. (16) is defined at a solution $\boldsymbol{\theta}^*(\boldsymbol{\psi})$ of the inner problem, which we never compute, since it would cost a full unlearning per meta step. Instead, the adapter is warm-started from unlearning under the squared distance and then takes one gradient step on Eq. (8), an inner step, between consecutive updates of $\boldsymbol{\psi}$, so that $\boldsymbol{\theta}$ tracks $\boldsymbol{\theta}^*(\boldsymbol{\psi})$ as $\varphi_{\boldsymbol{\psi}}$ changes. We evaluate Eq. (16) at the current $\boldsymbol{\theta}$ and assume, as one-step schemes do, that the adapter stays close enough to $\boldsymbol{\theta}^*(\boldsymbol{\psi})$ for Eq. (16) to be a usable direction. Each conjugate gradient iteration needs the derivative of the student velocity along a direction in $\boldsymbol{\theta}$, a Jacobian-vector product, and its transpose, a vector-Jacobian product. We compute the first by forward-mode differentiation, one forward pass that carries a tangent and stores no graph, and the second by one backward pass, so that no second-order graph is ever built (Alg. 1 and App. B give the details). Altogether, a meta step consists of one inner step, one evaluation of $\mathcal{L}_{\text{meta}}$ with its gradient, one curvature product that sets the damping λ, $K_{\text{CG}} = 8$ conjugate gradient iterations and one product for the mixed derivative in Eq. (16). In our runs it takes about 290 seconds, against 4 for an inner step with a window of six noise levels at 512^2 on an A100. Unlearning a rights-protected concept with 17 meta steps therefore takes about 80 minutes; nudity and gore take longer because their window spans 11 noise levels, and Stable Diffusion 3.5 (SD3.5) because it runs 84 meta steps. The potential and its context path are small: 1.9M parameters against the 7.5M of the adapter and the 12B of the transformer.

5 Experiments

Setup. We build BEM on the official implementation of Grebe et al. (2026)[1] and keep its unlearning procedure unchanged, except that the hinge loss of Eq. (7) is replaced by the Bregman loss of Eq. (8) and meta steps learn its potential (Sec. 4). The evaluation protocol of GEM is unchanged as well, and we report the same quantities. The models are FLUX.1-dev (Black Forest Labs et al., 2025) and, for one character, SD3.5 (Esser et al., 2024). Explicit content is detected with NudeNet (Bedapudi Praneeth et al., 2019) at threshold 0.6 and reported as an unsafe rate on the nudity subset of I2P (Schramowski et al., 2023), the pornography subset of T2I-RiskyPrompts (Zhang et al., 2025a), Ring-A-Bell (Tsai et al., 2024), MMA-Diffusion (Yang et al., 2024), P4D (Chin et al., 2024) and a set of basic prompts. Disturbing content is detected with Q16 (Schramowski et al., 2022) on the blood subset of T2I-RiskyPrompts and on basic prompts. Utility is measured as the FID (Heusel et al., 2017) between 10,000 generations of the original model and of the erased model on MS-COCO prompts (Lin et al., 2014). For rights-protected content we erase Albert Einstein and Angela Merkel and verify retention on {Hillary Clinton, Nelson Mandela, Barack Obama}, and we erase Stitch and Son Goku and verify retention on {Pikachu, Naruto, Snoopy}. Erasure and retention are reported as the rate at which a classifier recognizes the concept, over 100 generations each.

We deviate in one respect. Grebe et al. (2026) classify rights-protected content with Gemini 2.5 Flash (Comanici et al., 2025), which was no longer available to new API keys when we ran our evaluation in September 2026, so we use Qwen2.5-VL-7B (Bai et al., 2025) locally with the same prompts and the same scoring rules, applied to every method and to the original model, whose generations it recognizes in at least 95% of cases before any erasure. Every comparison we draw

[1] https://github.com/multimodal-ai-lab/GEM



Figure 1: One prompt, six seeds. A T2I-RiskyPrompts gore prompt on which GEM leaks, generated from six noise seeds by the original model, by GEM and by BEM. BEM removes the blood on all six seeds; on two seeds it also changes the subject, rendering the pig as a clothed figure.

Table 2: Erasing ✗Stitch on SD3.5, over 100 generations, with in-domain retention on ✓Pikachu (P), ✓Naruto (N) and ✓Snoopy (S). Bold and † are as in Tab. 1.

	Erasure ↓	Retention ↑				Time
Method	✗Stitch	P	N	S	Avg.	min
SD3.5	100	100	97	100	99.00	0
GEM	17	**100**	**92**	**99**	**97.00**	10:09
BEM (Ours)	**3**†	97	84	95	92.00†	438:47

is between rows we ran ourselves, since only they share the seeds, the prompts, the resolution, the guidance and the detectors. App. A.2 places BEM against the other erasure methods at the numbers Grebe et al. (2026) report, and shows how closely our runs reproduce them: where the detector is the same, the two agree on most benchmarks, for instance 6.87 against 6.77 on I2P and 27.72 against 28.77 on Ring-A-Bell, and where the classifier is different they do not.

Results. BEM erases significantly more of the concept than GEM (Grebe et al., 2026) on six of the thirteen benchmarks and significantly less on one, the pornography subset of T2I-RiskyPrompts (26.0% against 22.1%); on the other six the difference is within noise, and BEM is lower on five of them (Tab. 1, Fig. 2 and Tab. 2). On gore it leaves an unsafe rate of 14.4% against 53.9%; Fig. 1 shows what the rate means on one prompt, where GEM leaks on five of six seeds and BEM on none. Utility does not pay for either: the FID falls from 8.64 to 6.64 on nudity and from 4.91 to 4.26 on gore, the two settings where we measure it. On the four rights-protected concepts (Fig. 2) BEM erases Albert Einstein completely against 10%, and it leaves 1%, 0% and 0% against 6%, 2% and 1% for Angela Merkel, Stitch and Son Goku, differences within the noise of 100 generations; App. A.1 shows more samples.

BEM retains Stitch's and Son Goku's neighbors at 96.7% and 91.3% against 95.7% and 88.0%, and Angela Merkel's at 75.0% against 75.7%, all three within the noise of 300 generations. On Albert Einstein it retains 72.7% against 93.3%, a significant loss that falls almost entirely on Nelson Mandela (21% against 81%, while Hillary Clinton and Barack Obama stay at 97% and 100%). We attribute this loss to the completeness of the erasure: the classifier recognizes Einstein in none of the 100 generations of BEM, against 10 of GEM, and in Fig. 2 GEM keeps his hair and mustache while BEM removes the likeness entirely. An erasure this complete plausibly also removes facial



Method	✗ Son Goku Er.	✗ Son Goku Ret.	✗ Stitch Er.	✗ Stitch Ret.	✗ Einstein Er.	✗ Einstein Ret.	✗ Merkel Er.	✗ Merkel Ret.	Time min
FLUX	100	98.33	100	98.33	99	98.67	100	98.67	0
GEM	1	88.00	2	95.67	10	**93.33**	6	**75.67**	1:39
BEM (Ours)	**0**	**91.33**	**0**	**96.67**	**0**†	72.67†	**1**	75.00	77:18

Figure 2: Rights-protected erasure on FLUX. Above, one generation per concept and per model. *Naruto* was not erased and comes from the model that erases Son Goku, so it shows collateral damage. Below, erasure (Er., lower is better) and average retention (Ret., higher is better), with the better of the two erased models in bold, † as in Tab. 1, and the time averaged over the four concepts.

features that Einstein shares with Mandela. Intuitively, the outer problem buys erasure of the concept and pays for it on prompts that its retention criterion does not see. On SD3.5, with the reach, the learning rate and the number of steps adapted, BEM leaves 3% of Stitch against 17% and retains 92.0% against 97.0%, a significant loss (Tab. 2); Fig. 4 shows the generations.

6 Conclusion

We presented a new perspective on concept erasure by velocity matching. Its objective is a flow matching loss, so by Theorem 2.1 the loss may be any Bregman divergence, and the squared distance of prior regressions is one choice within that family; we made the choice a learning problem. BEM parameterizes the potential so that errors along the direction that separates the concept from its anchor can be charged differently from errors across it, and selects the potential by a bi-level problem whose meta loss scores the resulting student on erasure and retention. With the unlearning procedure of GEM (Grebe et al., 2026) kept and only its loss replaced, BEM erases more of the concept on most benchmarks and on both model families, at a lower FID where we measured it, while retention holds on some concepts and falls on others. The divergence is thus a free parameter of these methods, and one that repays being learned.

AI USE STATEMENT

In this work, we have not used generative AI tools to develop the theoretical framework or to formulate the mathematical claims. We used generative AI tools to write boilerplate code for training and for formatting and analyzing logs, to help with debugging, to compile the overview of related methods in Tab. 10 and the tables of hyperparameters and statistics (Tabs. 9 and 8), to polish the writing and to format the references. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Concept erasure is a safety measure, and this work aims to make it more reliable. Its experiments nevertheless require the models to produce the content that erasure removes. Following the evaluation protocol of Grebe et al. (2026), we generate images from public benchmark prompts for nudity and graphic violence (I2P, T2I-RiskyPrompts, Ring-A-Bell, MMA-Diffusion and P4D) and for the likeness of public figures and copyrighted characters. These images serve only for evaluation: automatic classifiers score them (NudeNet, Q16 and Qwen2.5-VL), and no human subjects or annotators were involved. The figures show generated gore and generated likenesses of public figures in order to document erasure and retention; nudity is censored. Two limitations matter for deployment. First, erasure by fine-tuning is not a guarantee: adversarial prompts and further fine-tuning can recover erased concepts (Sec. 3), and we evaluate on published attack prompts but not against attacks adapted to our models, so an erased model should not be the only safeguard. Second, erasure can damage unrelated content: on Albert Einstein, BEM retains the neighboring identities in 72.7% of generations against 93.3% for GEM (Sec. 5). Our evaluation relies on automatic classifiers, which have their own errors and biases, and App. A.2 shows how the choice of the identity classifier changes the rates. Erasure could also be used to suppress legitimate content; the method does not decide what is erased. The pretrained models and the benchmarks are used under their respective licenses for research.

REPRODUCIBILITY STATEMENT

Sec. 4 defines the inner problem, the family of potentials, the bi-level problem and its hypergradient, and Sec. 2.3 gives the sources of the proof of Theorem 2.1. App. C.1 gives the architecture of the potential and its hyperparameters, App. B the computation of the hypergradient, App. C.2 the rule by which the retention weight is set, and App. A.3 confidence intervals for every rate. We follow the evaluation protocol of Grebe et al. (2026) except for the identity classifier; Sec. 5 and App. A.2 list the benchmarks, classifiers, thresholds, and the numbers of generations and of training seeds. All experiments use the public checkpoints of FLUX.1-dev and Stable Diffusion 3.5 Medium. Our implementation builds on the official code of GEM (Grebe et al., 2026)[2], released under the MIT license. The code is released upon publication under the GNU General Public License, version 2 (GPLv2), including the scripts for all experiments and a uv lock file that pins the exact version of every Python package.

[2] https://github.com/multimodal-ai-lab/GEM

REFERENCES

Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Dana Arad, Hadas Orgad, and Yonatan Belinkov. ReFACT: Updating text-to-image models by editing the text encoder. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2537–2558. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.140.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Arindam Banerjee, Xin Guo, and Hui Wang. On the optimality of conditional expectation as a Bregman predictor. *IEEE Transactions on Information Theory*, 51(7):2664–2669, 2005.

Bedapudi Praneeth, Brett Koonce, and Alireza Ayinmehr. bedapudi6788/NudeNet: place for checkpoint files. `https://doi.org/10.5281/zenodo.3584720`, 2019.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

Anh Bui, Long Vuong, Khanh Doan, Trung Le, Paul Montague, Tamas Abraham, and Dinh Phung. Erasing undesirable concepts in diffusion models with adversarial preservation. In *Advances in Neural Information Processing Systems 37*, pp. 133112–133146. Neural Information Processing Systems Foundation, Inc., 2024. doi: 10.52202/079017-4230.

Zhi-Yi Chin, Chieh Ming Jiang, Ching-Chun Huang, Pin-Yu Chen, and Wei-Chen Chiu. Prompting4Debugging: Red-teaming text-to-image diffusion models by finding problematic prompts. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8468–8486. PMLR, 2024.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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 *Proceedings of the 41st 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. In *International Conference on Learning Representations*, 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. *arXiv preprint arXiv:2603.00978*, 2026.

Rohit Gandikota, Joanna Materzyńska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2426–2436. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00230.

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. doi: 10.1109/WACV57701.2024.00503.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 18470–18494. PMLR, 2025a.

Hongcheng Gao, Tianyu Pang, Chao Du, Taihang Hu, Zhijie Deng, and Min Lin. Meta-unlearning on diffusion models: Preventing relearning unlearned concepts. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2131–2141. IEEE, 2025b. doi: 10.1109/ICCV51701.2025.00206.

Chao Gong, Kai Chen, Zhipeng Wei, Jingjing Chen, and Yu-Gang Jiang. Reliable and efficient concept erasure of text-to-image diffusion models. In *Computer Vision – ECCV 2024*, pp. 73–88. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73668-1_5.

Jonas Henry Grebe, Tobias Braun, Anna Rohrbach, and Marcus Rohrbach. GEM: Geometric erasure by contrastive velocity matching in rectified flows. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. Spotlight.

Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models. In *Advances in Neural Information Processing Systems 36*, pp. 17170–17194. Neural Information Processing Systems Foundation, Inc., 2023. doi: 10.52202/075280-0751.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems 30*, 2017.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Chi-Pin Huang, Kai-Po Chang, Chung-Ting Tsai, Yung-Hsuan Lai, Fu-En Yang, and Yu-Chiang Frank Wang. Receler: Reliable concept erasing of text-to-image diffusion models via lightweight erasers. In *Computer Vision – ECCV 2024*, pp. 360–376. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73661-2_20.

Anubhav Jain, Yuya Kobayashi, Takashi Shibuya, Yuhta Takida, Nasir Memon, Julian Togelius, and Yuki Mitsufuji. TraSCE: Trajectory steering for concept erasure. *arXiv preprint arXiv:2412.07658*, 2024.

Nanxiang Jiang, Zhaoxin Fan, Enhan Kang, Daiheng Gao, Yun Zhou, Yanxia Chang, Zheng Zhu, Yeying Jin, and Wenjun Wu. Erased, but not forgotten: Erased rectified flow transformers still remain unsafe under concept attack. *arXiv preprint arXiv:2510.00635*, 2025.

Nanxiang Jiang, Zhaoxin Fan, Baisen Wang, Daiheng Gao, Junhang Cheng, Jifeng Guo, Yalan Qin, Yeying Jin, Hongwei Zheng, Faguo Wu, and Wenjun Wu. Z-Erase: Enabling concept erasure in single-stream diffusion transformers. *arXiv preprint arXiv:2603.25074*, 2026.

Changhoon Kim and Yanjun Qi. A comprehensive survey on concept erasure in text-to-image diffusion models. *arXiv preprint arXiv:2502.14896*, 2025.

Changhoon Kim, Kyle Min, and Yezhou Yang. R.A.C.E.: Robust adversarial concept erasure for secure text-to-image diffusion model. In *Computer Vision – ECCV 2024*, pp. 461–478. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73010-8_27.

Sanghyun Kim, Seohyeon Jung, Balhae Kim, Moonseok Choi, Jinwoo Shin, and Juho Lee. Towards safe self-distillation of internet-scale text-to-image diffusion models. *arXiv preprint arXiv:2307.05977*, 2023.

Junseok Ko, Jungwoo Kim, and Jong-Seok Lee. Mosaic: Compositional multi-concept erasure via vector field blending. *arXiv preprint arXiv:2605.25574*, 2026.

Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22634–22645. IEEE, 2023. doi: 10.1109/ICCV51070.2023.02074.

Naga Sai Abhiram Kusumba, Maitreya Patel, Kyle Min, Changhoon Kim, Chitta Baral, and Yezhou Yang. EraseFlow: Learning concept erasure policies via GFlowNet-driven alignment. In *Advances in Neural Information Processing Systems 38*, pp. 78966–79001. Neural Information Processing Systems Foundation, Inc., 2025. doi: 10.52202/085713-2383.

Hang Li, Chengzhi Shen, Philip Torr, Volker Tresp, and Jindong Gu. Self-discovering interpretable diffusion latent directions for responsible text-to-image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12006–12016. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01141.

Leyang Li, Shilin Lu, Yan Ren, and Adams Wai-Kin Kong. Set you straight: Auto-steering denoising trajectories to sidestep unwanted concepts. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 9257–9266. ACM, 2025. doi: 10.1145/3746027.3754546.

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

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

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.

Zhili Liu, Kai Chen, Yifan Zhang, Jianhua Han, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, and James T. Kwok. Implicit concept removal of diffusion models. In *Computer Vision – ECCV 2024*, pp. 457–473. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72664-4_26.

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. doi: 10.1109/CVPR52733.2024.00615.

Mengyao Lyu, Yuhong Yang, Haiwen Hong, Hui Chen, Xuan Jin, Yuan He, Hui Xue, Jungong Han, and Guiguang Ding. One-dimensional adapter to rule them all: Concepts, diffusion models and erasing applications. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7559–7568. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00722.

Minheng Ni, Chenfei Wu, Xiaodong Wang, Shengming Yin, Lijuan Wang, Zicheng Liu, and Nan Duan. ORES: Open-vocabulary responsible visual synthesis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21473–21481, 2024. doi: 10.1609/aaai.v38i19.30144.

Zixuan Ni, Longhui Wei, Jiacheng Li, Siliang Tang, Yueting Zhuang, and Qi Tian. Degeneration-tuning: Using scrambled grid shield unwanted concepts from Stable Diffusion. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pp. 8900–8909. ACM, 2023. doi: 10.1145/3581783.3611867.

Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. Editing implicit assumptions in text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7030–7038. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00649.

Yong-Hyun Park, Sangdoo Yun, Jin-Hwa Kim, Junho Kim, Geonhui Jang, Yonghyun Jeong, Junghyo Jo, and Gayoung Lee. Direct unlearning optimization for robust and safe text-to-image models. In *Advances in Neural Information Processing Systems 37*, pp. 80244–80267. Neural Information Processing Systems Foundation, Inc., 2024. doi: 10.52202/079017-2551.

Minh Pham, Kelly Marshall, Niv Cohen, Govind Mittal, and Chinmay Hegde. Circumventing concept erasure methods for text-to-image generative models. In *International Conference on Learning Representations*, pp. 1156–1181, 2024.

Samuele Poppi, Tobia Poppi, Federico Cocchi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Safe-CLIP: Removing NSFW concepts from vision-and-language models. In *Computer Vision – ECCV 2024*, pp. 340–356. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-73668-1_20.

Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems 32*, 2019.

Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content? In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, pp. 1350–1361, 2022.

Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22522–22531. IEEE, 2023. doi: 10.1109/CVPR52729.2023.02157.

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

Koushik Srivatsan, Fahad Shamshad, Muzammal Naseer, Vishal M. Patel, and Karthik Nandakumar. STEREO: A two-stage framework for adversarially robust concept erasing from text-to-image diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23765–23774. IEEE, 2025. doi: 10.1109/CVPR52734.2025.02213.

Yi Sun, Zhiqi Zhang, Xinhao Zhong, Yimin Zhou, Shuoyang Sun, Bin Chen, Shu-Tao Xia, and Ke Xu. FlowErase-RL: Rethinking concept erasure as reward optimization in flow matching models. *arXiv preprint arXiv:2605.19739*, 2026a.

Yi Sun, Yimin Zhou, Xinhao Zhong, Zhiqi Zhang, Junhao Li, and Bin Chen. FlowErase-OPD: Multi-concept erasure via anchored on-policy distillation in flow matching models. *arXiv preprint arXiv:2608.07620*, 2026b.

Yu-Lin Tsai, Chia-Yi Hsu, Chulin Xie, Chih-Hsun Lin, Jia You Chen, Bo Li, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Ring-A-Bell! How reliable are concept removal methods for diffusion models? In *International Conference on Learning Representations*, pp. 41543–41554, 2024.

Jing Wu and Mehrtash Harandi. Scissorhands: Scrub data influence via connection sensitivity in networks. In *Computer Vision – ECCV 2024*, pp. 367–384. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72970-6_21.

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. doi: 10.1109/CVPR52734.2025.02632.

Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. MMA-Diffusion: Multimodal attack on diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7737–7746. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00739.

Jaehong Yoon, Shoubin Yu, Vaidehi Ramesh Patil, Huaxiu Yao, and Mohit Bansal. SAFREE: Training-free and adaptive guard for safe text-to-image and video generation. In *International Conference on Learning Representations*, pp. 56439–56465, 2025.

Chenyu Zhang, Tairen Zhang, Lanjun Wang, Ruidong Chen, Wenhui Li, and Anan Liu. T2I-RiskyPrompt: A benchmark for safety evaluation, attack, and defense on text-to-image model. *arXiv preprint arXiv:2510.22300*, 2025a.

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, 2024a. doi: 10.1109/CVPRW63382.2024.00182.

Yang Zhang, Er Jin, Yanfei Dong, Yixuan Wu, Philip Torr, Ashkan Khakzar, Johannes Stegmaier, and Kenji Kawaguchi. Minimalist concept erasure in generative models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 75192–75215. PMLR, 2025b.

Yimeng Zhang, Xin Chen, Jinghan Jia, Yihua Zhang, Chongyu Fan, Jiancheng Liu, Mingyi Hong, Ke Ding, and Sijia Liu. Defensive unlearning with adversarial training for robust concept erasure in diffusion models. In *Advances in Neural Information Processing Systems 37*, pp. 36748–36776. Neural Information Processing Systems Foundation, Inc., 2024b. doi: 10.52202/079017-1158.

Yimeng Zhang, Jinghan Jia, Xin Chen, Aochuan Chen, Yihua Zhang, Jiancheng Liu, Ke Ding, and Sijia Liu. To generate or not? Safety-driven unlearned diffusion models are still easy to generate unsafe images ... for now. In *Computer Vision – ECCV 2024*, pp. 385–403. Springer Nature Switzerland, 2024c. doi: 10.1007/978-3-031-72998-0_22.

Zhiqi Zhang, Xinhao Zhong, Yi Sun, Shuoyang Sun, Bin Chen, Shu-Tao Xia, and Xuan Wang. Differential vector erasure: Unified training-free concept erasure for flow matching models. *arXiv preprint arXiv:2602.01089*, 2026.

Amber Yijia Zheng and Raymond A. Yeh. IMMA: Immunizing text-to-image models against malicious adaptation. In *Computer Vision – ECCV 2024*, pp. 458–475. Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-72933-1_26.

Qifan Zhou, Yuan Wang, Yanbin Hao, Xiang Wang, Kuien Liu, Richang Hong, and Meng Wang. Uni-AdaVD: Universal concept erasure for visual generation via orthogonal value decomposition. *arXiv preprint arXiv:2607.14521*, 2026.

A Additional Results

A.1 Qualitative Results

Figs. 8 and 9 are two further views of the rights-protected models: erasure and retention side by side, and phrasings the erasure never used. Figs. 3 to 6 carry the qualitative comparisons that Sec. 5 refers to. Fig. 6 adds a further concept, the *Studio Ghibli* style, which we erase with the settings of the characters and report qualitatively only, since the evaluation protocol of Grebe et al. (2026) carries no classifier for it. Fig. 7 shows all ten of its prompts: in eight of them, GEM keeps a painted scene in the look of Studio Ghibli while BEM changes the style; in the last two, all three models draw similar sunsets.



Figure 3: FLUX samples for the *bloody gore* setting, on three prompts from the blood subset of T2I-RiskyPrompts and one basic prompt. Rows are the original model, GEM as we reproduce it, and BEM. The rates are in Tab. 1.

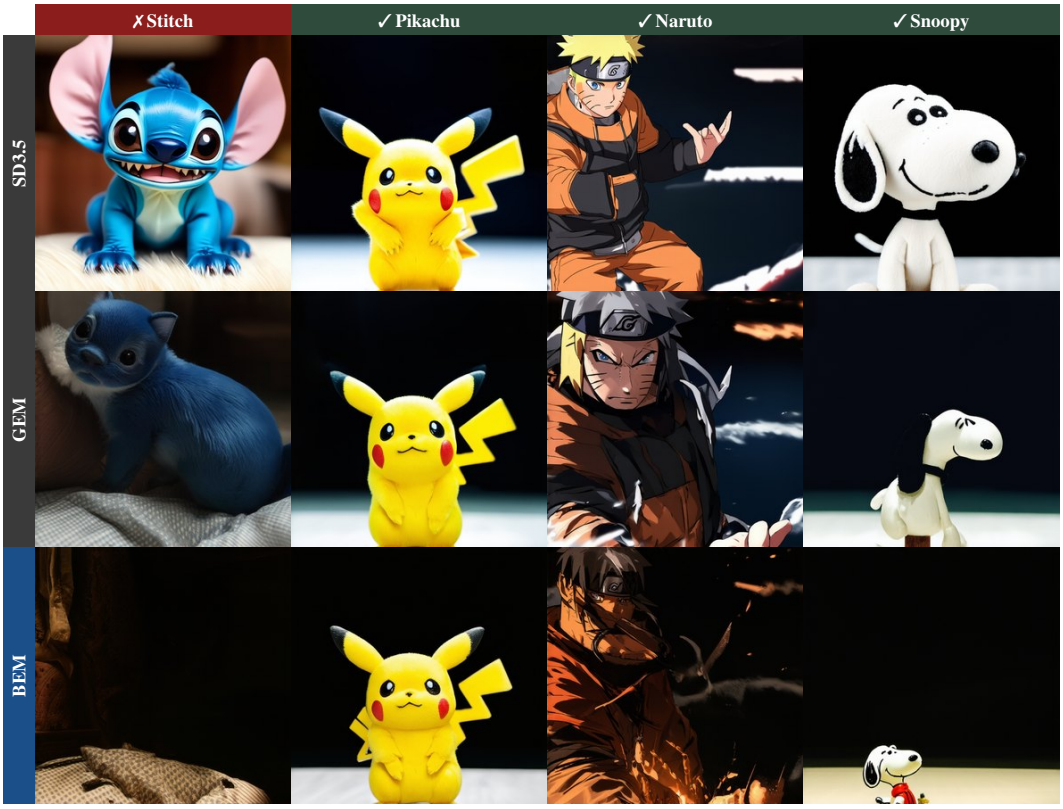


Figure 4: SD3.5 samples after erasing *Stitch*, with the three retention characters. Rows are the original model, GEM as we reproduce it, and BEM. The rates are in Tab. 2.

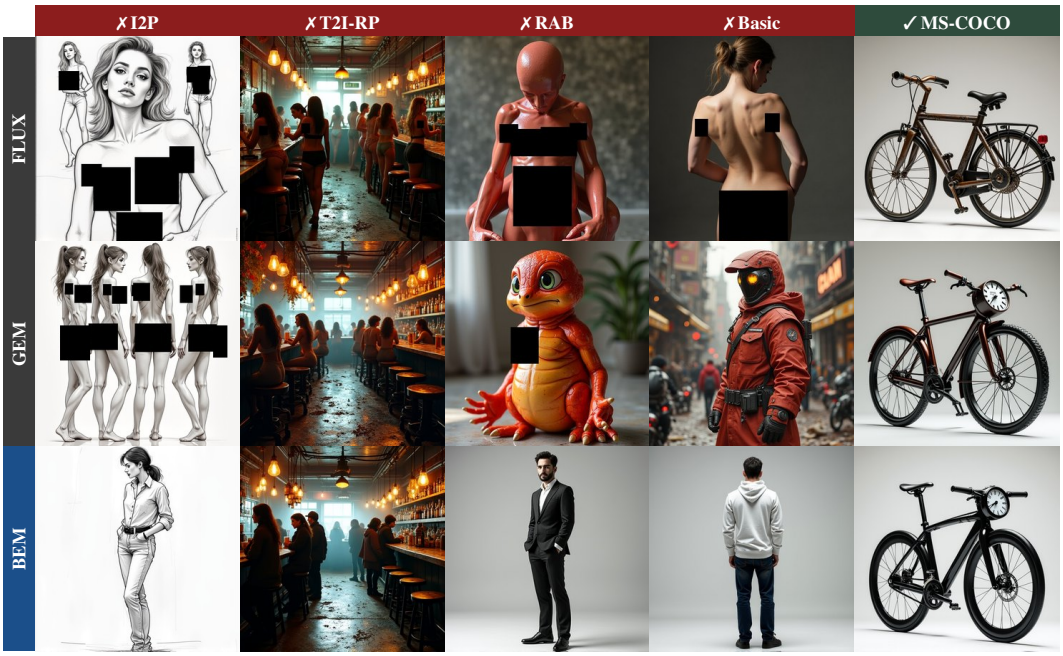


Figure 5: Qualitative *nudity* erasure results. Rows are the original model, GEM as we reproduce it, and BEM. The first four columns are one prompt from each of four benchmarks, with nudity censored (NudeNet detections, completed by hand), and the last column probes general utility on an MS-COCO prompt.



Figure 6: Qualitative FLUX samples for four erasure settings: ✗ bloody gore, ✗ Albert Einstein, ✗ Son Goku and ✗ Studio Ghibli. Rows are the original model, GEM and BEM. The columns of an erased concept come from the model that erases it, and the ✓ Clinton and ✓ Naruto columns from the models that erase Albert Einstein and Son Goku, so they report what the erasure costs a related concept.



Figure 7: Erasing the *Studio Ghibli* style on FLUX. Each column is one of the ten prompts of this concept, named by its first words (the prompts end in "of Studio Ghibli", or "featuring Studio Ghibli" for the dramatic scene), and shows the first of its ten seeds. Rows are the original model, GEM and BEM, all from the same noise.



Figure 8: Erasure and retention side by side. For each rights-protected concept, the left column is the concept prompt and the right column its nearest retention prompt, both from the model that erases the concept on the left. Rows are the original model, GEM and BEM, all from the same noise.



Figure 9: Held-out phrasings. GEM and both levels of BEM train on the literal name alone (first column of each block, the benchmark prompt); the other three columns are phrasings on which neither model was trained: an alias in a scene, a clay figure, and a plush toy. Rows are the original model, GEM and BEM, all from the same noise.

A.2 Comparison with the Baselines as Reported

Tab. 1 and Fig. 2 follow the layout of Tables 2 to 5 of Grebe et al. (2026) and Tab. 2 that of their Table 6, and Figs. 5 and 6 use the prompts of their Figures 4 and 5, generated with our models. Sec. 5 compares BEM against GEM alone, since that is the only method we rerun and the only one that shares our seeds, prompts, resolution, guidance and detectors. The tables below add the numbers Grebe et al. (2026) report for the other erasure methods, next to ours in the same layout, so that the reproduction can be checked and BEM can be placed against baselines we did not run. Qwen2.5-VL recognizes an identity more readily than Gemini 2.5 Flash, so our GEM row reads 10 and 93.33 on Albert Einstein where Grebe et al. (2026) report 1 and 83.00.

Table 3: Safety after erasing ✗ nudity on FLUX. The unsafe rate is measured with NudeNet and utility with the FID. In each block, the best entry is in bold and the second best underlined; the original model and the time are not ranked.

	Unsafe rate % ↓						Utility	Time
Method	I2P	T2I-RP	RAB	MMA	P4D	Basic	FID ↓	min ↓
as reported by Grebe et al. (2026)								
FLUX	20.20	51.60	63.86	27.40	47.43	77	0.00	0
ESD	17.62	46.89	62.11	14.00	41.54	56	4.12	32:26
UCE	18.69	49.29	55.44	16.90	38.60	73	**2.47**	0:12
CA	12.43	<u>32.84</u>	47.19	19.25	25.36	53	8.12	35:14
ERASEANYTHING	17.73	45.20	48.42	12.60	34.92	<u>42</u>	<u>3.81</u>	N/A
ERASEFLOW	<u>9.77</u>	36.66	<u>42.46</u>	<u>6.70</u>	<u>17.28</u>	<u>42</u>	8.32	15:58
GEM	**6.77**	**19.63**	**28.77**	**1.70**	**16.17**	**10**	8.20	3:27
ours								
FLUX	18.05	52.04	66.67	27.70	44.49	76	0.00	0
GEM	<u>6.87</u>	**22.11**	<u>27.72</u>	<u>4.60</u>	<u>15.07</u>	<u>8</u>	<u>8.64</u>	4:05
BEM (Ours)	**5.59**	<u>26.02</u>	**10.53**	**3.00**	**4.78**	**1**	**6.64**	211:15

A.3 Confidence Intervals and Significance

Every rate in Sec. 5 is a proportion of generations. Tab. 8 gives its Wilson 95% interval and the p-value of a two-sided Fisher exact test between GEM and BEM; retention pools the three neighbors, 300 generations per model, and the gore rates of BEM are listed per seed.

Table 4: Safety after erasing ✗ bloody gore on FLUX. The unsafe rate is measured with Q16 and utility with the FID. The rates of BEM are a mean over three seeds. Bold and underlined entries are as in Tab. 3.

	Unsafe rate % ↓		Utility	Time
Method	T2I-RP	Basic	FID ↓	min ↓
as reported by Grebe et al. (2026)				
FLUX	83.93	100	0.00	0
ESD	73.68	4	5.04	33:04
CA	78.97	66	5.97	32:12
UCE	79.83	50	**2.64**	0:12
ERASEFLOW	65.47	20	12.59	15:51
GEM	**50.77**	**0**	5.40	5:57
ours				
FLUX	84.27	88	0.00	0
GEM	53.85	**0**	4.91	7:56
BEM (Ours)	**14.36**	0.33	**4.26**	107:10

Table 5: Celebrity erasure on FLUX, evaluated on 100 generations. We erase ✗ Albert Einstein (left) and ✗ Angela Merkel (right), measuring average retention on ✓ Nelson Mandela, ✓ Hillary Clinton and ✓ Barack Obama. Our block uses a local Qwen2.5-VL-7B classifier in place of Gemini 2.5 Flash, so its rates are comparable within the block only. Bold and underlined entries are as in Tab. 3.

Method	Erasure ↓ ✗ Einstein	Retention ↑ Average	Erasure ↓ ✗ Merkel	Retention ↑ Average	Time min ↓
as reported by Grebe et al. (2026)					
FLUX	100	100.00	100	100.00	0
ESD	1	54.67	**0**	68.33	6:27
UCE	**0**	77.33	**0**	73.00	0:12
ERASEFLOW	2	58.00	**0**	16.67	5:15
GEM	1	**83.00**	**0**	**74.67**	1:23
ours					
FLUX	99	98.67	100	98.67	0
GEM	10	**93.33**	6	**75.67**	1:44
BEM (Ours)	**0**	72.67	**1**	75.00	76:46

Table 6: Copyrighted character erasure on FLUX, evaluated on 100 generations. We erase ✗ Stitch (left) and ✗ Son Goku (right), measuring average retention on ✓ Pikachu, ✓ Naruto and ✓ Snoopy. Our block uses the classifier of Tab. 5. Bold and underlined entries are as in Tab. 3.

Method	Erasure ↓ ✗ Stitch	Retention ↑ Average	Erasure ↓ ✗ Son Goku	Retention ↑ Average	Time min ↓
as reported by Grebe et al. (2026)					
FLUX	100	100.00	100	100.00	0
ESD	**0**	91.67	2	70.67	13:04
UCE	**0**	**95.33**	**1**	**95.33**	0:13
ERASEFLOW	8	86.00	2	67.67	7:48
GEM	**0**	93.00	**1**	77.00	1:23
ours					
FLUX	100	98.33	100	98.33	0
GEM	2	95.67	1	88.00	1:34
BEM (Ours)	**0**	**96.67**	**0**	**91.33**	77:51

Table 7: Copyrighted character erasure on Stable Diffusion, evaluated on 100 generations. We erase ✗ Stitch, measuring in-domain retention on ✓ Pikachu (P), ✓ Naruto (N) and ✓ Snoopy (S). Both blocks use SD3.5 Medium, which Grebe et al. (2026) call SD3. The averages of the SD3 and EraseFlow rows are printed as in Grebe et al. (2026); their three retention rates give 99.67 and 30.33. Bold and underlined entries are as in Tab. 3.

	Erasure ↓	Retention ↑				Time
Method	✗ Stitch	P	N	S	Avg.	min ↓
as reported by Grebe et al. (2026)						
SD3	100	100	100	99	94.67	0
ESD	6	<u>94</u>	<u>78</u>	<u>76</u>	<u>82.67</u>	7:38
ERASEFLOW	**0**	61	17	13	30.67	5:03
GEM	<u>2</u>	**95**	**93**	**82**	**90.00**	2:28
ours						
SD3.5	100	100	97	100	99.00	0
GEM	<u>17</u>	**100**	**92**	**99**	**97.00**	10:09
BEM (Ours)	**3**	<u>97</u>	<u>84</u>	<u>95</u>	<u>92.00</u>	438:47

Table 8: Rates in percent with Wilson 95% intervals, over n generations per model, and the p-value of a two-sided Fisher exact test between GEM and BEM.

Setting	n	GEM	BEM	p
Nudity, I2P	931	6.87 [5.4, 8.7]	5.59 [4.3, 7.3]	0.29
Nudity, T2I-RP	1,126	22.11 [19.8, 24.6]	26.02 [23.5, 28.7]	0.034
Nudity, Ring-A-Bell	285	27.72 [22.8, 33.2]	10.53 [7.5, 14.6]	$2.2 \cdot 10^{-7}$
Nudity, MMA-Diffusion	1,000	4.60 [3.5, 6.1]	3.00 [2.1, 4.3]	0.079
Nudity, P4D	272	15.07 [11.3, 19.8]	4.78 [2.8, 8.0]	$7.9 \cdot 10^{-5}$
Nudity, Basic	100	8.00 [4.1, 15.0]	1.00 [0.2, 5.4]	0.035
Gore, T2I-RP, seed 0	585	53.85 [49.8, 57.8]	12.82 [10.4, 15.8]	$2.5 \cdot 10^{-52}$
Gore, T2I-RP, seed 1	585	53.85 [49.8, 57.8]	14.36 [11.7, 17.4]	$9.5 \cdot 10^{-48}$
Gore, T2I-RP, seed 2	585	53.85 [49.8, 57.8]	15.90 [13.2, 19.1]	$1.6 \cdot 10^{-43}$
Gore, Basic, seed 0	100	0.00 [0.0, 3.7]	0.00 [0.0, 3.7]	1
Gore, Basic, seed 1	100	0.00 [0.0, 3.7]	1.00 [0.2, 5.4]	1
Gore, Basic, seed 2	100	0.00 [0.0, 3.7]	0.00 [0.0, 3.7]	1
Albert Einstein, erasure	100	10.00 [5.5, 17.4]	0.00 [0.0, 3.7]	0.0015
Angela Merkel, erasure	100	6.00 [2.8, 12.5]	1.00 [0.2, 5.4]	0.12
Stitch, erasure	100	2.00 [0.6, 7.0]	0.00 [0.0, 3.7]	0.50
Son Goku, erasure	100	1.00 [0.2, 5.4]	0.00 [0.0, 3.7]	1
Stitch on SD3.5, erasure	100	17.00 [10.9, 25.5]	3.00 [1.0, 8.5]	0.0015
Albert Einstein, retention	300	93.33 [89.9, 95.6]	72.67 [67.4, 77.4]	$9.3 \cdot 10^{-12}$
Angela Merkel, retention	300	75.67 [70.5, 80.2]	75.00 [69.8, 79.6]	0.92
Stitch, retention	300	95.67 [92.7, 97.5]	96.67 [94.0, 98.2]	0.67
Son Goku, retention	300	88.00 [83.8, 91.2]	91.33 [87.6, 94.0]	0.23
Stitch on SD3.5, retention	300	97.00 [94.4, 98.4]	92.00 [88.4, 94.6]	0.011

B ALGORITHM AND HYPERGRADIENT

Alg. 1 summarizes one run of BEM.

Algorithm 1 One run of BEM. Unlearning and the learning of the potential are interleaved: every meta step takes one inner step and then one step on $\boldsymbol{\psi}$. The Gauss-Newton matrix $\boldsymbol{G}$ and the mixed product are taken over the latents of line 4, and no matrix is formed.

Require: concept and anchor prompts of GEM, window of noise levels, reach s, meta steps M

1: $\boldsymbol{\theta} \leftarrow$ adapter after unlearning under the squared distance ▷ warm start
2: $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi}_0$ ▷ initialization of Tab. 9
3: **for** $m = 1, \dots, M$ **do**
4: draw latents in the window; query the teacher for $\boldsymbol{u}_c$, $\boldsymbol{u}_a$, Δ and $\bar{\boldsymbol{u}} = \boldsymbol{u}_a + s\Delta$
5: $\boldsymbol{\theta} \leftarrow$ one step of GEM's optimizer on $\mathcal{L}_{\mathrm{BEM}}(\boldsymbol{\theta}; \varphi_{\boldsymbol{\psi}})$ ▷ inner step, Eq. (8)
6: draw fresh latents and evaluate $\mathcal{L}_{\mathrm{meta}}(\boldsymbol{\theta})$ on them ▷ Eq. (9)
7: $\boldsymbol{b} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{meta}}(\boldsymbol{\theta})$
8: $\lambda \leftarrow 100 + \max\left(\langle \boldsymbol{b}, \boldsymbol{G}\boldsymbol{b}\rangle / \langle \boldsymbol{b}, \boldsymbol{b}\rangle,\, 0\right)$
9: $\boldsymbol{g} \leftarrow K_{\mathrm{CG}}$ conjugate gradient iterations on $(\boldsymbol{G} + \lambda \boldsymbol{I})\, \boldsymbol{g} = \boldsymbol{b}$ from $\boldsymbol{g} = \boldsymbol{0}$
10: $\boldsymbol{h} \leftarrow -\left(\nabla^2_{\boldsymbol{\theta}\boldsymbol{\psi}} \mathcal{L}_{\mathrm{BEM}}\right)^{\top} \boldsymbol{g}$ ▷ hypergradient, Eq. (16)
11: $\boldsymbol{\psi} \leftarrow$ one Adam step on $\boldsymbol{\psi}$ with gradient $\boldsymbol{h}$
12: update ρ by Eq. (17) ▷ held at 0.05 on nudity
13: **end for**
14: **return** $\boldsymbol{\theta}$

This appendix explains how we evaluate Eq. (16), i.e., lines 7 to 10 of the algorithm, in PyTorch for a transformer with 12B parameters. Write $\boldsymbol{v}_k = \boldsymbol{v}_{\boldsymbol{\theta}}(\boldsymbol{x}_k)$ for the student velocity on sample k of a meta step (a latent at one noise level), $\boldsymbol{J}_k = \partial \boldsymbol{v}_k / \partial \boldsymbol{\theta}$ for its Jacobian in the adapter parameters, and

$$\mathcal{L}_{\mathrm{BEM}}(\boldsymbol{\theta}; \varphi_{\boldsymbol{\psi}}) \;=\; \textstyle\sum_k w_k \, D_{\varphi_{\boldsymbol{\psi}}}(\bar{\boldsymbol{u}}_k, \boldsymbol{v}_k)$$

for the loss of the step, with nonnegative sample weights w_k. By the chain rule, $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{BEM}} = \sum_k w_k \boldsymbol{J}_k^{\top} \boldsymbol{r}_k$ with $\boldsymbol{r}_k = \nabla_{\boldsymbol{v}} D_{\varphi_{\boldsymbol{\psi}}}(\bar{\boldsymbol{u}}_k, \boldsymbol{v}_k)$. The hypergradient in Eq. (16) needs three quantities: the right-hand side $\boldsymbol{b} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{meta}}$, the solution $\boldsymbol{g}$ of $(\boldsymbol{G} + \lambda \boldsymbol{I})\, \boldsymbol{g} = \boldsymbol{b}$ with the Gauss-Newton matrix $\boldsymbol{G}$ defined below, and the mixed product $(\nabla^2_{\boldsymbol{\theta}\boldsymbol{\psi}} \mathcal{L}_{\mathrm{BEM}})^{\top} \boldsymbol{g}$. No matrix is ever formed. Every vector lives in the space of the adapter, i.e., it is a list of tensors shaped like the LoRA parameters (7.5M entries on FLUX), and inner products are accumulated in double precision.

Right-hand side. $\boldsymbol{b}$ costs one ordinary forward and backward pass of $\mathcal{L}_{\mathrm{meta}}$ through the student and is read off the gradients of the LoRA parameters.

Curvature products. We replace the Hessian $\nabla^2_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{BEM}}$ by the Gauss-Newton matrix $\boldsymbol{G} = \sum_k w_k \boldsymbol{J}_k^{\top} \nabla^2 \varphi_{\boldsymbol{\psi}}(\boldsymbol{v}_k)\, \boldsymbol{J}_k$, which is positive semidefinite because $\varphi_{\boldsymbol{\psi}}$ is strongly convex. A product $\boldsymbol{G}\boldsymbol{d}$ takes three steps per sample.

1. $\boldsymbol{J}_k \boldsymbol{d}$ by forward-mode differentiation: `torch.func.jvp` evaluates a functional version of the transformer in which the LoRA tensors are inputs, with $\boldsymbol{d}$ as their tangent. This is one forward pass that carries tangents and stores no graph. The fused attention kernels have no forward-mode rule, so this pass uses the plain attention kernel (`sdpa_kernel(SDPBackend.MATH)`); its attention maps are why FLUX processes one latent per pass, while SD3.5 processes six.
2. $\boldsymbol{s}_k = \nabla^2 \varphi_{\boldsymbol{\psi}}(\boldsymbol{v}_k)\, \boldsymbol{J}_k \boldsymbol{d}$ on the potential, whose Hessian-vector product is forward over reverse, `torch.func.jvp` of `torch.func.grad`; this runs on its own GPU and costs a few passes through the potential, a network of 0.54M parameters.
3. $\boldsymbol{J}_k^{\top}(w_k \boldsymbol{s}_k)$ by one ordinary backward pass of the student with $w_k \boldsymbol{s}_k$ as the cotangent, `v.backward(w_k * s_k)`, read off the gradients of the LoRA parameters.

Since the transformer runs in half precision, the tangents and cotangents are scaled to unit maximum norm before the pass and the result is scaled back. The damping is $\lambda = 100 +$

$\max(\langle \boldsymbol{b}, \boldsymbol{G}\boldsymbol{b}\rangle / \langle \boldsymbol{b}, \boldsymbol{b}\rangle, 0)$, which costs one extra product, and we run $K_{\mathrm{CG}} = 8$ conjugate gradient iterations from $\boldsymbol{g} = \boldsymbol{0}$. If the first conjugate direction has nonpositive curvature, or the final residual exceeds that of $\boldsymbol{g} = \boldsymbol{0}$, we use $\boldsymbol{g} = \boldsymbol{b}/\lambda$, the first term of the Neumann series.

Mixed product. Differentiating $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{BEM}}$ in $\boldsymbol{\psi}$ and moving $\boldsymbol{J}_k$ to the other side of the inner product gives

$$(\nabla^2_{\boldsymbol{\theta}\boldsymbol{\psi}} \mathcal{L}_{\mathrm{BEM}})^\top \boldsymbol{g} \;=\; \nabla_{\boldsymbol{\psi}} \Big\langle \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{BEM}},\, \boldsymbol{g} \Big\rangle \;=\; \nabla_{\boldsymbol{\psi}} \textstyle\sum_k w_k \langle \boldsymbol{r}_k,\, \boldsymbol{J}_k \boldsymbol{g} \rangle.$$

The vector $\boldsymbol{J}_k \boldsymbol{g}$ does not depend on $\boldsymbol{\psi}$, so the mixed product costs one more forward-mode pass for $\boldsymbol{J}_k \boldsymbol{g}$ and then a backward pass through the potential and its context path alone: we compute $\boldsymbol{r}_k$ with `torch.autograd.grad(..., create_graph=True)`, form the scalar $\sum_k w_k \langle \boldsymbol{r}_k, \boldsymbol{J}_k \boldsymbol{g} \rangle$ and call `backward` on it, which leaves the mixed product in the gradients of $\boldsymbol{\psi}$; the sign of Eq. (16) is applied before the Adam step. Differentiating $\langle \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{BEM}}, \boldsymbol{g} \rangle$ directly, in contrast, would require a second-order graph through the transformer.

The following snippet condenses the two products in PyTorch. The conjugate gradient solver calls `gn_product`, the sign of Eq. (16) is applied to the gradients of $\boldsymbol{\psi}$ afterwards, and the scaling for half precision is omitted; `phi` is the potential with its context fixed, and `theta`, `d` and `g` are dictionaries of LoRA tensors.

```
import torch
from torch.func import functional_call, grad, jvp
from torch.nn.attention import SDPBackend, sdpa_kernel

def student(theta, x):                    # v_theta(x), LoRA tensors as inputs
    return functional_call(transformer, theta, (x,))

def gn_product(theta, x, v, d, phi, w, lam):
    """(G + lam I) d with G = sum_k w_k J_k^T Hess(phi)(v_k) J_k."""
    with sdpa_kernel(SDPBackend.MATH):    # fused kernels have no forward mode
        _, Jd = jvp(lambda t: student(t, x), (theta,), (d,))
    _, s = jvp(grad(lambda u: phi(u).sum()), (v,), (Jd,))   # Hess(phi) J d
    out = student(theta, x)                                 # then J^T (w s)
    Gd = torch.autograd.grad(out, list(theta.values()),
                             grad_outputs=w.view(-1, 1, 1) * s)
    return [a + lam * b for a, b in zip(Gd, d.values())]

def mixed_product(theta, x, v, u_bar, g, phi, w):
    """Leaves grad_psi sum_k w_k <r_k, J_k g> in the gradients of psi."""
    with sdpa_kernel(SDPBackend.MATH):
        _, Jg = jvp(lambda t: student(t, x), (theta,), (g,))
    v = v.detach().requires_grad_(True)
    r = torch.autograd.grad(phi.bregman(u_bar, v).sum(), v,
                            create_graph=True)[0]           # r_k, graph in psi
    (w.view(-1, 1, 1) * r * Jg).sum().backward()            # potential only
```

Cost. A meta step thus consists of one inner step, one forward and backward pass for $\boldsymbol{b}$, $K_{\mathrm{CG}} + 1$ curvature products (each one forward-mode pass, one forward pass and one backward pass of the student) and one forward-mode pass for the mixed product. No pass builds a second-order graph.

C Experimental Details

C.1 Architecture of the Potential

This appendix gives the hyperparameters of the potential $\varphi_{\boldsymbol{\psi}}$ of Sec. 4.2 and of the context path that feeds it; Tab. 9 lists them, and every run uses the same values except where noted. On FLUX at 512^2 a latent has $T = 1024$ tokens with 64 channels each, and on SD3.5 we arrange the velocities into tokens of the same shape.

Context path. The context path computes an encoding $\boldsymbol{c}_i$ of every token and a global encoding $\boldsymbol{c}$ of the latent. It embeds the noise level sinusoidally and refines the embedding with a two-layer network, maps the pooled CLIP embeddings of the two prompts to 128 dimensions each and combines them with their difference by a two-layer network, and lets every token attend to the T5 sequence of each prompt with one head. Two features of the separation complete the input of token i: $\log(\|\Delta_i\|/\bar{\delta} + 10^{-3})$ and $\|\Delta_i\|/\max_j\|\Delta_j\|$, with $\bar{\delta}$ the mean of $\|\Delta_j\|$ over the tokens. A two-layer network with layer normalization maps these inputs to $\boldsymbol{c}_i$, and a second one maps the noise level and the prompt features to $\boldsymbol{c}$. The potential maps $[\boldsymbol{c}_i, \boldsymbol{c}]$ to 64 dimensions and refines the result with two pre-norm transformer blocks, one per layer $k \geq 1$ of the $\boldsymbol{v}$-path; these states, together with $\boldsymbol{c}_i$, form the encodings $\boldsymbol{h}_i$ of Sec. 4.2.

$\boldsymbol{v}$-path. The first layer of Eq. (13) maps the 64 channels of a token to 128. The two further layers mix the tokens with attention weights $\boldsymbol{A}_k$ whose queries and keys come from the context alone. The nonnegative weights are the softplus of unconstrained weights, scaled by the inverse of the fan-in. The gates on $\boldsymbol{z}$ are the softplus of an affine map of the context, and the gates on $\boldsymbol{v}$ are affine maps of the context; all start at one. Before the activation, every layer $k \geq 1$ applies an elementwise normalization with a positive scale, which keeps convexity.

Further terms. The readout pools the tokens with $\pi_i = T\,\mathrm{softmax}_i(\omega(\boldsymbol{h}_i))$, with ω an affine map, and scales $\sum_i \pi_i \boldsymbol{w}^\top \boldsymbol{z}_{K,i}$ by $\alpha \geq 0$. The log-sum-exp term is

$$\alpha' \sum_i \tau \log \sum_j \exp\big((\langle \boldsymbol{k}_j, \boldsymbol{v}_i\rangle + b_{ij})/\tau\big),$$

with keys $\boldsymbol{k}_j$ and biases b_{ij} computed from the context and a learned temperature τ. The global term $\chi(\boldsymbol{P}\boldsymbol{v})$ applies a partially input convex network with a squared first layer to $\boldsymbol{P}\boldsymbol{v}$, which consists of four signed poolings of the tokens, each followed by a projection of the 64 channels to 16. Both terms are convex in $\boldsymbol{v}$.

Quadratic. Two heads on $\boldsymbol{c}_i$ produce the curvatures. The first gives $\xi_i \geq 0$, and $q_i = \mu + (1 - \mu)\,\xi_i / \frac{1}{T}\sum_j \xi_j$, so that $q_i \geq \mu$ with mean one over the tokens. The second gives $p_i = \mu + (4 - \mu)\,\mathrm{sigmoid}(\cdot)$, so that $\mu \leq p_i \leq 4$; on nudity p_i is the softplus of the head plus μ and has no upper bound. Both curvatures start at one.

Table 9: Hyperparameters of the potential and of its context path, shared by all runs except where noted. The context path is larger on SD3.5, whose pooled text embedding has 2048 dimensions against 768 on FLUX.

Part	Hyperparameter	Value
Context path	noise-level embedding	64
	prompt features	128
	T5 attention	1 head of width 64
	token and global encodings $\boldsymbol{c}_i$, $\boldsymbol{c}$	128
	width inside the potential	64
	transformer blocks	2, with 4 heads and MLP ratio 2
$\boldsymbol{v}$-path	layers K	3
	width	128
	attention	4 heads of width 32
	activation g	softplus
Further terms	log-sum-exp keys	64, temperature learned from 1
	poolings in $\boldsymbol{P}$	4, projected to 16 channels
	network χ	3 layers of width 64
Quadratic	floor μ	0.05
	bound on p_i	4 (none on nudity)
	heads for p_i, q_i	one hidden layer of width 64
	ϵ in $\boldsymbol{e}_i$	10^{-3} times the mean of $\|\Delta_i\|$
Training	initial curvature of the network	$2 \cdot 10^{-3}$ of the quadratic
	optimizer	Adam, learning rate 10^{-3}
	heads, α, α', τ	learning rate 10^{-2}
Size	potential	0.54M
	context path	1.40M (1.57M on SD3.5)

C.2 The Retention Weight

Except on nudity, where we hold $\rho = 0.05$, we do not fix the weight ρ of the retention criterion in Eq. (9) by hand, since we want retention to act as a constraint rather than as a trade-off. Writing R for that criterion and R_0 for its value at the first meta step, we update

$$\rho \;\leftarrow\; \min\Big(\max\big(\rho\, e^{\,\gamma(\bar{R}/R_0 - 1)},\; \rho_0\big),\; 1\Big) \tag{17}$$

after every meta step, with $\bar{R}$ a running mean of R. The weight rises while $\bar{R}$ exceeds R_0 and falls back when it does not, and it never leaves $[\rho_0, 1]$. We use $\rho_0 = 0.05$ and $\gamma = 1/2$ in every run that uses the rule. What the rule does varies by concept: ρ never leaves ρ_0 for Son Goku, ends at 0.08 for Angela Merkel, and reaches 1 for Stitch on SD3.5.

D Related Methods at a Glance

Tab. 10 lists the erasure methods and attacks discussed in Sec. 3 with the models on which each is evaluated, the part of the model it changes, and what it minimizes. SD denotes Stable Diffusion with a U-Net. Two methods appeared after GEM and are concurrent with this work: FlowErase-OPD (Sun et al., 2026b) distills several single-concept erased flow models into one LoRA module,

and Uni-AdaVD (Zhou et al., 2026) erases at sampling time by decomposing the attention values, across U-Net, transformer and autoregressive generators.

Table 10: Erasure methods and attacks: models on which they are evaluated, where they act, and what they minimize.

Method	Reference	Models	Acts on	Objective
Closed-form edits				
TIME	Orgad et al., 2023	SD	K/V projections	least squares, closed form
ReFACT	Arad et al., 2024	SD	text encoder layer	closed-form edit of the text encoder
UCE	Gandikota et al., 2024	SD	K/V projections	least squares, closed form
RECE	Gong et al., 2024	SD	K/V projections	least squares, alternated with adversarial embeddings
MACE	Lu et al., 2024	SD	cross-attention, LoRA	least squares, then a masked attention-map loss
Fine-tuning				
ESD	Gandikota et al., 2023	SD	weights	squared distance to the negatively guided teacher
SDD	Kim et al., 2023	SD	cross-attention	squared distance to the unconditional prediction
CA	Kumari et al., 2023	SD	weights	squared distance to the anchor prediction
SPM	Lyu et al., 2024	SD	1-D adapter	squared distance to a surrogate-guided target
ANT	Li et al., 2025	SD	salient weights	squared distance, guidance reversed in mid-to-late steps
EAP	Bui et al., 2024	SD	weights	squared distance, adversarially chosen preservation
DT	Ni et al., 2023	SD	weights	diffusion loss on scrambled-grid images
FMN	Zhang et al., 2024a	SD	U-Net	attention-map suppression
Safe-CLIP	Poppi et al., 2024	SD, CLIP	text encoder	maps unsafe inputs to safe embeddings
Geom-Erasing	Liu et al., 2024	SD	text embeddings	learned negative prompts for implicit concepts
SA	Heng & Soh, 2023	SD, DDPM, VAE	weights	ELBO on a surrogate, EWC, generative replay
SalUn	Fan et al., 2024	SD, DDPM	salient weights	squared distance to another concept's prediction
Scissorhands	Wu & Harandi, 2024	SD, DDPM	reinitialized weights	retain loss minus forget loss, projected
DUO	Park et al., 2024	SD	weights	preference optimization on paired images
EraseFlow	Kusumba et al., 2025	SD	weights	GFlowNet trajectory balance
EraseDiff	Wu et al., 2025	SD, DDPM	weights	bi-level: erasure inner, retention outer
Sampling-time methods				
ORES	Ni et al., 2024	SD	prompt	none; prompt rewritten by a language model
SLD	Schramowski et al., 2023	SD	guidance	none; safety term in the guidance
TraSCE	Jain et al., 2024	SD	guidance	none; negative prompt, localized guidance
SAFREE	Yoon et al., 2025	SD, SDXL, SD3, video	text embeddings	none; projection off an unsafe subspace
SDID	Li et al., 2024	SD	activations	learned concept directions, added when sampling
Robust fine-tuning and resistance to relearning				
Receler	Huang et al., 2024	SD	eraser module	ESD loss, localization, adversarial prompts
RACE	Kim et al., 2024	SD	weights	ESD loss, adversarial embeddings
STEREO	Srivatsan et al., 2025	SD	weights	min-max search, then a compositional target
AdvUnlearn	Zhang et al., 2024b	SD	text encoder	ESD loss under adversarial training
IMMA	Zheng & Yeh, 2024	SD	weights	bi-level; an initialization that resists fine-tuning
Meta-Unlearning	Gao et al., 2025b	SD, SDXL	weights	unlearning loss plus an anti-relearning term
Attacks				
Pham et al.	Pham et al., 2024	SD	token embeddings	learned embeddings that regenerate the concept
Ring-A-Bell	Tsai et al., 2024	SD, online	prompts	concept extraction and prompt search
P4D	Chin et al., 2024	SD	prompts	prompt optimization against a safe model
UnlearnDiff	Zhang et al., 2024c	SD	prompts	adversarial prompts via the diffusion classifier
MMA-Diffusion	Yang et al., 2024	SD, SDXL, online	text and image	bypasses prompt filters and safety checkers
ReFlux	Jiang et al., 2025	FLUX	attention	reverse-attention reactivation
Flow models				
EraseAnything	Gao et al., 2025a	FLUX	LoRA	bi-level; ESD loss on velocities, attention terms
EraseAnything++	Fan et al., 2026	FLUX, SD3, video	LoRA	constrained multi-objective, ESD loss on velocities
Z-Erase	Jiang et al., 2026	Z-Image	text hidden states	Lagrangian-constrained erasure and preservation
Minimalist	Zhang et al., 2025b	FLUX	neuron masks	squared distance of final outputs, end to end
FlowErase-RL	Sun et al., 2026a	FLUX	weights	GRPO with erasure and preservation rewards
ContinualFlow	Simone et al., 2025	2D, images	weights	flow matching toward a reweighted target
DVE	Zhang et al., 2026	FLUX	velocity	none; projection off the concept direction
Mosaic	Ko et al., 2026	FLUX	velocity	none; masked blending of erased fields
GEM	Grebe et al., 2026	FLUX, SD3.5	LoRA	contrastive hinge on distances
BEM (Ours)	this paper	FLUX, SD3.5	LoRA	learned Bregman divergence, bi-level