EraseDiff: Erasing Data Influence in Diffusion Models

Jing Wu
Monash University
jing.wu1monash.edu

Trung Le
Monash University
trunglm@monash.edu

Munawar Hayat
Monash University
munawar.hayat@monash.edu

Mehrtash Harandi
Monash University
mehrtash.harandi@monash.edu

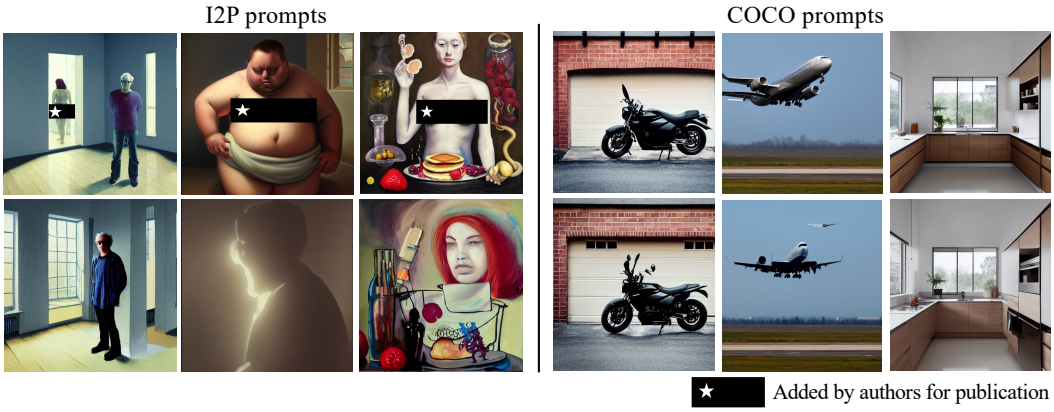


Figure 1: Top to Bottom: generated samples by SD v1.4 and model scrubbed by our method, *EraseDiff*, when erasing the concept of 'nudity'. *EraseDiff* can avoid NSFW content while preserving model utility. Source code is available at https://github.com/JingWu321/EraseDiff.

Abstract

We introduce EraseDiff, an unlearning algorithm designed for diffusion models to address concerns related to data memorization. Our approach formulates the unlearning task as a constrained optimization problem, aiming to preserve the utility of the diffusion model on retained data while removing the information associated with the data to be forgotten. This is achieved by altering the generative process to deviate away from the ground-truth denoising procedure. To manage the computational complexity inherent in the diffusion process, we develop a first-order method for solving the optimization problem, which has shown empirical benefits. Extensive experiments and thorough comparisons with state-of-the-art algorithms demonstrate that EraseDiff effectively preserves the model's utility, efficacy, and efficiency.

WARNING: This paper contains sexually explicit imagery that may be offensive in nature.

1 Introduction

Diffusion Models (Ho et al., 2020; Song et al., 2020; Rombach et al., 2022) are now the method of choice in deep generative models, owing to their high-quality output, stability, and ease of training

Preprint. Under review.

procedure. This has facilitated their successful integration into commercial applications such as *midjourney*. Unfortunately, the ease of use associated with diffusion models brings forth significant privacy risks. Studies have shown that these models can memorize and regenerate individual images from their training datasets (Somepalli et al., 2023a,b; Carlini et al., 2023). Beyond privacy, diffusion models are susceptible to misuse and can generate inappropriate digital content (Rando et al., 2022; Salman et al., 2023; Schramowski et al., 2023). They are also vulnerable to poison attacks (Chen et al., 2023b), allowing the generation of target images with specific triggers. These factors collectively pose substantial security threats. Moreover, the ability of diffusion models to emulate distinct artistic styles (Shan et al., 2023; Gandikota et al., 2023a) raises questions about data ownership and compliance with intellectual property and copyright laws.

In this context, individuals whose images are used for training might request the removal of their private data. In particular, data protection regulations like the European Union General Data Protection Regulation (GDPR) (Voigt & Von dem Bussche, 2017) and the California Consumer Privacy Act (CCPA) (Goldman, 2020) grant users the *right to be forgotten*, obligating companies to expunge data pertaining to a user upon receiving a request for deletion. These legal provisions grant data owners the right to remove their data from trained models and eliminate its influence on said models (Bourtoule et al., 2021; Guo et al., 2020; Golatkar et al., 2020; Mehta et al., 2022; Sekhari et al., 2021; Ye et al., 2022; Tarun et al., 2023b,a; Chen et al., 2023a).

A straightforward solution for unlearning is to retrain the model from scratch after excluding the data that needs to be forgotten. However, the removal of pertinent data followed by retraining diffusion models from scratch demands substantial resources and is often deemed impractical. A version of the stable diffusion model trained on subsets of the LAION-5B dataset (Schuhmann et al., 2022) costs approximately 150,000 GPU hours with 256 A100 GPUs[1]. Existing research on efficient unlearning have primarily focused on classification problems (Karasuyama & Takeuchi, 2010; Cao & Yang, 2015; Ginart et al., 2019; Bourtoule et al., 2021; Wu et al., 2020; Guo et al., 2020; Golatkar et al., 2020; Mehta et al., 2022; Sekhari et al., 2021; Chen et al., 2023a). Despite substantial progress, methods developed for unlearning in classification are observed to be ineffective for generation tasks as studied by Fan et al. (2023). Consequently, there is a pressing need for the development of methods capable of scrubbing data from diffusion models without necessitating complete retraining.

Recently, a handful of studies (Gandikota et al., 2023a,b; Zhang et al., 2023; Heng & Soh, 2023a,b; Fan et al., 2023) target unlearning in diffusion models, with a primary focus on the text-to-image models (Gandikota et al., 2023a,b; Zhang et al., 2023; Bui et al., 2024). Heng & Soh (2023b) utilize ideas from continual learning to preserve model utility when performing forgetting for a wide range of generative models. Their method requires the computation of the Fisher Information Matrix (FIM) for different datasets and models, which could lead to significant computational demands. Fan et al. (2023) propose to shift the attention to salient weights w.r.t. the forgetting data, resulting in a very potent unlearning algorithm across image classification and generation tasks.

In this work, we propose *EraseDiff*, and formulate diffusion unlearning as a constrained Optimization problem, where the objective is to finetune the models with the remaining data $\mathcal{D}_r$ for preserving the model utility and to erase the influence of the forgetting data $\mathcal{D}_f$ on the models by deviating the learnable reverse process from the ground-truth denoising procedure, namely minimizing the loss over the remaining data while maximizing that over the forgetting data. A common issue in unlearning is the gradient conflict, as optimizing one objective could hinder another one. To address this issue, we adopt an approximate optimization problem that identifies an optimal direction to update different objectives. We benchmark *EraseDiff* on various scenarios, encompassing unlearning of classes on CIFAR10 (Krizhevsky et al., 2009) with Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020), classes on Imagenette (Howard & Gugger, 2020) and concepts on the I2P dataset (Schramowski et al., 2023) with stable diffusion. Our empirical findings show that *EraseDiff* is $11\times$ faster than Heng and Soh's method (Heng & Soh, 2023b) and $2\times$ faster than Fan's method (Fan et al., 2023) when forgetting on DDPM while achieving better unlearning results across several metrics. The results demonstrate that *EraseDiff* is capable of effectively erasing data influence in diffusion models, ranging from specific classes to the concept of nudity.

[1]https://stablediffusion.gitbook.io/overview/stable-diffusion-overview/technology/training-procedures

2 Background

In this section, we outline the components of the models we evaluate, including DDPM, classifier-free guidance diffusion models (Ho & Salimans, 2022), Latent Diffusion Models (LDM) (Rombach et al., 2022). Throughout the paper, we denote scalars, and vectors/matrices by lowercase and bold symbols, respectively (e.g., a, $\boldsymbol{a}$, $\boldsymbol{A}$).

DDPM. (1) Diffusion: DDPM gradually diffuses the data distribution $\mathbb{R}^d \ni \mathbf{x}_0 \sim q(\mathbf{x})$ into the standard Gaussian distribution $\mathbb{R}^d \ni \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ with T time steps, ie., $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}_d)$, where $\alpha_t = 1 - \beta_t$ and $\{\beta_t\}_{t=1}^T$ are the pre-defined variance schedule. The diffusion takes the form $\mathbf{x}_t$ as $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, where $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. (2) Training: A model $\epsilon_{\boldsymbol{\theta}}(\cdot)$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ is trained to learn the reverse process $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx q(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Given $\mathbf{x}_0 \sim q(\mathbf{x})$ and time step $t \in [1, T]$, the simplified training objective is to minimize the distance between $\boldsymbol{\epsilon}$ and the predicted $\boldsymbol{\epsilon}_t$ given $\mathbf{x}_0$ at time t, ie., $\|\boldsymbol{\epsilon} - \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\|$. (3) Sampling: after training the model, we could obtain the learnable backward distribution $p_{\boldsymbol{\theta}^*}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t))$, where $\boldsymbol{\mu}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t, t))$ and $\boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}(\mathbf{x}_t, t) = \frac{(1-\bar{\alpha}_{t-1})\beta_t}{1-\bar{\alpha}_t}$. Then, given $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, $\mathbf{x}_0$ could be obtained via sampling from $p_{\boldsymbol{\theta}^*}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ from $t = T$ to $t = 1$ step by step.

Classifier-free guidance. Classifier-free guidance is a conditioning method to guide diffusion-based generative models without an external pre-trained classifier. Model prediction would be $\epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c, t)$, where c is the input's corresponding label. The unconditional and conditional models are jointly trained by randomly setting c to the unconditional class $\emptyset$ identifier with the probability p_{uncond}. Then, the sampling procedure uses the linear combination of the conditional and unconditional score estimates as $\boldsymbol{\epsilon}_t = (1+w) \cdot \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c) - w \cdot \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|\emptyset)$, w is the guidance scale that controls the strength of the classifier guidance.

Latent diffusion model. Latent diffusion models (LDM) apply the diffusion models in the latent space $\mathbf{z}$ of a pre-trained variational autoencoder. The noise would be added to $\mathbf{z} = \varepsilon(\mathbf{x})$, instead of the data $\mathbf{x}$, and the denoised output would be transformed to image space with the decoder. Besides, cross-attention layers are introduced into the model for general conditioning inputs.

3 Diffusion Unlearning

Let $\mathcal{D} = \{\mathbf{x}_i, c_i\}_i^N$ be a dataset of images $\mathbf{x}_i$ associated with label c_i representing the class. $\mathcal{C} = \{1, \cdots, C\}$ denotes the label space where C is the total number of classes and $c_i \in \mathcal{C}$. We split the training data $\mathcal{D}$ into the forgetting data $\mathcal{D}_f \subset \mathcal{D}$ and its complement, remaining data $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$. The forgetting data has label space $\mathcal{C}_f \subseteq \mathcal{C}$, and the remaining label space is denoted as $\mathcal{C}_r = \mathcal{C} \setminus \mathcal{C}_f$.

3.1 Training objective

Our goal is to scrub the information about $\mathcal{D}_f$ carried by the diffusion models while maintaining the model utility over the remaining data $\mathcal{D}_r$. To achieve this, we adopt different training objectives for $\mathcal{D}_r$ and $\mathcal{D}_f$ as follows.

For the remaining data $\mathcal{D}_r$, we fine-tune the diffusion models with the original objective:

$$\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r) = \mathbb{E}_{t, \boldsymbol{\epsilon} \in \mathcal{N}(\mathbf{0}, \mathbf{I}_d), (\mathbf{x}_0, c) \sim \mathcal{D}_r \times \mathcal{C}_r}[\|\boldsymbol{\epsilon} - \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c)\|_2^2], \tag{1}$$

where $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. For the forgetting data $\mathcal{D}_f$, we aim to let the models fail to generate meaningful images corresponding to $\mathcal{C}_f$ and thus propose:

$$\mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f) = \mathbb{E}_{t, \boldsymbol{\epsilon} \in \mathcal{N}(\mathbf{0}, \mathbf{I}_d), (\mathbf{x}_0, c) \sim \mathcal{D}_f \times \mathcal{C}_f}[\|\boldsymbol{\epsilon}_f - \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c)\|_2^2]. \tag{2}$$

With this, we hinder the approximator $\epsilon_{\boldsymbol{\theta}}$ to guide the denoising process to obtain meaningful examples for the forgetting data example $\mathbf{x}_0 \sim \mathcal{D}_f$. In our experiments, we choose $\boldsymbol{\epsilon}_f$ to be a distribution different from $\boldsymbol{\epsilon} \in \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. This could be $\boldsymbol{\epsilon}_f \in \mathcal{U}(\mathbf{0}, \mathbf{I}_d)$, or $\epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c_m)$ like Fan et al. (2023); Heng & Soh (2023b) where $c_m \neq c$ so that the denoised image $\mathbf{x}_0$ is not related to the forgetting class/concept c.

To perform unlearning and minimize $\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r)$ and $\mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f)$ simultaneously, it is common to form

$$\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r) + \lambda \mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f), \tag{3}$$

with $\lambda \geq 0$ as the optimization objective (see for example Fan et al. (2023)). However, training could be hindered due to the conflicting gradients between the retaining and forgetting objectives.

Equation (3) could also be viewed as a scalarization of a Multi-Objective Optimization (MOO) problem, ie., minimizing $\left(\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r), \mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f)\right)^\top$. It is well known that MOO should address the gradient conflict issue.

Instead of scalarization of MOO, we propose to minimize the following objective:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \ \mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r) \\ &\text{s.t.} \min_{\boldsymbol{\phi}} \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f, \boldsymbol{\phi}_{\texttt{init}} = \boldsymbol{\theta}) \ . \end{aligned} \tag{4}$$

Here, the problem $\min_{\boldsymbol{\phi}} \ \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f, \boldsymbol{\phi}_{\texttt{init}} = \boldsymbol{\theta})$ indicates that given $\boldsymbol{\theta}$, the optimization of $\boldsymbol{\phi}$ starts from $\boldsymbol{\theta}$ and aims to minimize the forgetting loss. In other words, if optimality $\boldsymbol{\theta}^*$ is achieved, we have found $\boldsymbol{\theta}^*$ that maintains the model's utility as a result of $\min_{\boldsymbol{\theta}} \ \mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r)$, and starting from $\boldsymbol{\theta}^*$, we cannot further reduce the forgetting loss due to $\min_{\boldsymbol{\phi}} \ \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f, \boldsymbol{\phi}_{\texttt{init}} = \boldsymbol{\theta}^*)$. Now, we note that $\arg\min_{\boldsymbol{\theta}} \{\mathcal{L}(\boldsymbol{\theta}) - \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \leq 0\} \in \{\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})\}$. This insight will aid us in solving Equation (4) efficiently as we will show next. Putting everything together, we propose:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \ \mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r) \\ \text{s.t.} \quad &\mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f) - \min_{\boldsymbol{\phi}} \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f, \boldsymbol{\phi}_{\texttt{init}} = \boldsymbol{\theta}) \leq 0, \end{aligned} \tag{5}$$

where $\boldsymbol{\phi}$ is initialized at $\boldsymbol{\theta}$.

3.2 Solution

To solve Equation (5), let us first denote $g(\boldsymbol{\theta}) = \mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f) - \min_{\boldsymbol{\phi}} \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f)$. Suppose that the current solution for Equation (5) is $\boldsymbol{\theta}_t$, we aim to update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{\delta}_t$ where η is sufficiently small, so that $\mathcal{L}_r(\boldsymbol{\theta}_{t+1}; \mathcal{D}_r)$ decreases (ie., preserve model utility) and $g(\boldsymbol{\theta}_{t+1})$ decreases (ie., erasure). To this end, inspired by Liu et al. (2022), we aim to find $\boldsymbol{\delta}_t$ by:

$$\begin{aligned} &\boldsymbol{\delta}_t \in \frac{1}{2} \operatorname{argmin}_{\boldsymbol{\delta}} \left\| \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r) - \boldsymbol{\delta} \right\|_2^2, \\ \text{s.t.} \quad &\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^\top \boldsymbol{\delta} \geq a_t > 0. \end{aligned} \tag{6}$$

This will ensure that the update $\boldsymbol{\delta}_t$ is close to $\nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r)$ and decreases $g(\boldsymbol{\theta}_t)$ until it reaches stationary. Because $g\left(\boldsymbol{\theta}_{t+1}\right) - g\left(\boldsymbol{\theta}_t\right) \approx -\eta \nabla_{\boldsymbol{\theta}} g\left(\boldsymbol{\theta}_t\right)^\top \boldsymbol{\delta} \leq -\eta a_t < 0$, we can ensure that $g\left(\boldsymbol{\theta}_{t+1}\right) < g\left(\boldsymbol{\theta}_t\right)$ for small step size $\eta > 0$. This means that the update $\boldsymbol{\delta}_t$ can ensure to minimize $\mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f)$ as long as it does not conflict with descent of $\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r)$.

To find the solution to the optimization problem in Equation (6), the following theorem is developed:

Theorem 3.1. *The optimal solution of the optimization problem in Equation* (6) *is* $\boldsymbol{\delta}^* = \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r) + \lambda_t \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)$ *where* $\lambda_t = \max\{0, \frac{a_t \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r)}{\|\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)\|_2^2}\}$.

Proof. The Lagrange function with $\lambda \geq 0$ for Equation (6):

$$h(\boldsymbol{\delta}, \lambda) = \frac{1}{2} \left\| \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t; \mathcal{D}_r) - \boldsymbol{\delta} \right\|_2^2 + \lambda (a_t - \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^\top \boldsymbol{\delta}). \tag{7}$$

Then, using the Karush-Kuhn-Tucker (KKT) theorem, at the optimal solution we have

$$\begin{aligned} \boldsymbol{\delta} - \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r) - \lambda \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t) &= \mathbf{0}, \\ \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^\top \boldsymbol{\delta} &\geq a_t, \\ \lambda (a_t - \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^T \boldsymbol{\delta}) &= 0, \\ \lambda &\geq 0. \end{aligned} \tag{8}$$

From the above constraints, we can obtain:

$$\begin{aligned} \boldsymbol{\delta} &= \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r) + \lambda \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t), \\ \lambda &= \max\{0, \frac{a_t - \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r)}{\|\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)\|_2^2}\}. \end{aligned} \tag{9}$$

□

Algorithm 1 *EraseDiff*: Erasing Data Influence in Diffusion Models.

Input: Well-trained model with parameters $\boldsymbol{\theta}_0$, forgetting data $\mathcal{D}_f$ and remaining data $\mathcal{D}_r$, outer iteration number T and inner iteration number K, learning rate η.

Output: Parameters $\boldsymbol{\theta}^*$ for the scrubbed model.

1: **for** iteration t in T **do**
2: $\quad\boldsymbol{\phi}^0 = \boldsymbol{\theta}_t$.
3: $\quad$Get $\boldsymbol{\phi}^K$ by K steps of gradient descent on $\mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f)$ starting from $\boldsymbol{\phi}^0$.
4: $\quad$Set $g(\boldsymbol{\theta}_t) = \mathcal{L}_f(\boldsymbol{\theta}_t; \mathcal{D}_f) - \mathcal{L}_f(\boldsymbol{\phi}^K; \mathcal{D}_f)$.
5: $\quad$Update the model: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta(\nabla_{\boldsymbol{\theta}_t}\mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r) + \lambda_t \nabla_{\boldsymbol{\theta}_t} g(\boldsymbol{\theta}_t; \boldsymbol{\phi}^K))$,
6: $\quad$where $\lambda_t = \max\{0, \frac{a_t - \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^T \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r)}{\|\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)\|_2^2}\}$.
7: **end for**

In practice, we can choose $a_t = \eta\|\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)\|_2^2$. The remaining question is how to compute $\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)$. For this computation, we start from $\boldsymbol{\phi}^0 = \boldsymbol{\theta}_t$ and use gradient descend in K steps with the learning rate $\xi > 0$ to reach $\boldsymbol{\phi}^K$, namely $\boldsymbol{\phi}^{k+1} = \boldsymbol{\phi}^k - \xi \nabla_{\boldsymbol{\phi}} \mathcal{L}_f(\boldsymbol{\phi}^k; \mathcal{D}_f)$ and $k = 0, \cdots, K-1$. Finally, we can compute the update $\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t) = \nabla_{\boldsymbol{\theta}} \mathcal{L}_f(\boldsymbol{\theta}_t; \mathcal{D}_f) - \nabla_{\boldsymbol{\phi}^K} \mathcal{L}_f(\boldsymbol{\phi}^K; \mathcal{D}_f)$.

We can characterize the solution of our algorithm as follows:

Theorem 3.2 (Pareto optimality)**.** *The stationary point obtained by our algorithm is Pareto optimal of the problem* $\min_{\boldsymbol{\theta}}[\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r), \mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f)]$.

Proof. Let $\boldsymbol{\theta}^*$ be the solution to our problem. Recall that for the current $\boldsymbol{\theta}$, we find $\boldsymbol{\phi}^K$ to minimize $g(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f) - \min \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f)$. Assume that we can update in sufficient number of steps K so that $\boldsymbol{\phi}^K = \boldsymbol{\phi}^*(\boldsymbol{\theta}) = \operatorname{argmin}_{\boldsymbol{\phi}} g(\boldsymbol{\theta}, \boldsymbol{\phi}) = \operatorname{argmin}_{\boldsymbol{\phi}} \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f)$. Here $\boldsymbol{\phi}$ is initialized at $\boldsymbol{\theta}$.

The objective aims to minimize $\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r) + \lambda g(\boldsymbol{\theta}; \boldsymbol{\phi}^*(\boldsymbol{\theta}))$, let $\boldsymbol{\theta}^*$ be the optimal solution to this objective. Note that $g(\boldsymbol{\theta}, \boldsymbol{\phi}^*(\boldsymbol{\theta})) = \mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f) - \min \mathcal{L}_f(\boldsymbol{\phi}^*(\boldsymbol{\theta}); \mathcal{D}_f) \geq 0$ as $\boldsymbol{\phi}$ starts from $\boldsymbol{\theta}$ and is update to decreas $\mathcal{L}_m(\boldsymbol{\phi}; \mathcal{D}_f)$. This will decrease to 0 for minimizing the above objective. Therefore, at the optimal solution $\boldsymbol{\theta}^*$, we have $g(\boldsymbol{\theta}^*, \boldsymbol{\phi}^*(\boldsymbol{\theta}^*)) = 0$. This further implies that $\mathcal{L}_f(\boldsymbol{\theta}^*; \mathcal{D}_f) = \min \mathcal{L}_f(\boldsymbol{\phi}^*(\boldsymbol{\theta}^*); \mathcal{D}_f)$, meaning that $\boldsymbol{\theta}^*$ is the current optimal solution of $\mathcal{L}_f(\boldsymbol{\theta}; D_f)$ because we cannot update further the optimal solution. Moreover, we have $\boldsymbol{\theta}^*$ as the local minima of $\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r)$ in sufficiently small vicinity considered, because in the small vicinity around $\boldsymbol{\theta}^*$, $g(\boldsymbol{\theta}, \boldsymbol{\phi}^*(\boldsymbol{\theta}^*)) = 0$ provides no further improvements for the above sum, any increase in the above objective in the vicinity of $\boldsymbol{\theta}^*$ would primarily be due to an increase in $\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r)$.

□

4 Related Work

Memorization. Privacy of generative models has been studied extensively for GANs (Feng et al., 2021; Meehan et al., 2020; Webster et al., 2021) and generative language models (Carlini et al., 2022, 2021; Jagielski et al., 2022; Tirumala et al., 2022). These generative models often risk replicating from their training data. Recently, several studies (Carlini et al., 2023; Somepalli et al., 2023b,a; Vyas et al., 2023) investigated these data replication behaviors in diffusion models, raising concerns about the privacy and copyright issues. Possible mitigation strategies are deduplicating and randomizing conditional information (Somepalli et al., 2023b,a), or training models with differential privacy (DP) (Abadi et al., 2016; Dwork et al., 2006; Dwork, 2008; Dockhorn et al., 2022). However, leveraging DP-SGD (Abadi et al., 2016) may cause training to diverge (Carlini et al., 2023).

Malicious misuse. Diffusion models usually use training data from varied open sources and when such unfiltered data is employed, there is a risk of it being tainted (Chen et al., 2023b) or manipulated (Rando et al., 2022), resulting in inappropriate generation (Schramowski et al., 2023). They also risk the imitation of copyrighted content, e.g., mimicking the artistic style (Gandikota et al., 2023a; Shan et al., 2023). To counter inappropriate generation, data censoring (Gandhi et al., 2020; Birhane & Prabhu, 2021; Nichol et al., 2021; Schramowski et al., 2022) where excluding black-listed images before training, and safety guidance where diffusion models will be updated away from the inappropriate/undesired concept (Gandikota et al., 2023a; Schramowski et al., 2023) are proposed.

Shan et al. (2023) propose protecting artistic style by adding barely perceptible perturbations to the artworks before public release. Yet, Rando et al. (2022) argue that DMs can still generate content that bypasses the filter. Chen et al. (2023b) highlight the susceptibility of DMs to poison attacks, where target images are generated with specific triggers.

Machine unlearning. Removing data directly involves retraining the model from scratch, which is inefficient and impractical. Thus, to reduce the computational overhead, efficient machines unlearning methods (Romero et al., 2007; Karasuyama & Takeuchi, 2010; Cao & Yang, 2015; Ginart et al., 2019; Bourtoule et al., 2021; Wu et al., 2020; Guo et al., 2020; Golatkar et al., 2020; Mehta et al., 2022; Sekhari et al., 2021; Chen et al., 2023a; Tarun et al., 2023b) have been proposed. Several studies (Gandikota et al., 2023a,b; Heng & Soh, 2023a,b; Fan et al., 2023; Zhang et al., 2023; Bui et al., 2024) recently introduce unlearning in diffusion models. Most of them (Gandikota et al., 2023a,b; Heng & Soh, 2023a; Zhang et al., 2023) mainly focus on text-to-image models and high-level visual concept erasure. Heng & Soh (2023b) adopt Elastic Weight Consolidation (EWC) and Generative Replay (GR) from continual learning to perform unlearning effectively without access to the training data. Heng and Soh's method can be applied to a wide range of generative models, however, it needs the computation of FIM for different datasets and models, which may lead to significant computational demands. Fan et al. (2023) propose a very potent unlearning algorithm called SalUn that shifts attention to important parameters w.r.t. the forgetting data. SalUn can perform effectively across image classification and generation tasks.

In this work, we introduce a simple yet effective unlearning algorithm for diffusion models by formulating the problem as a constrained optimization problem. Below, we will show that our algorithm is not only faster than Heng and Soh's method (Heng & Soh, 2023b) and Fan's method (Fan et al., 2023), but even outperforms these methods in terms of the trade-off between the forgetting and preserving model utility.

5 Experiment

We evaluate *EraseDiff* in various scenarios, including removing images with specific classes/concepts, to answer the following research questions (RQs): (i) Can typical machine unlearning methods be applied to diffusion models? (ii) Is the proposed method able to remove the influence of $\mathcal{D}_f$ in the diffusion models? (iii) Is the proposed method able to preserve the model utility while removing $\mathcal{D}_f$? (iv) Is the proposed method efficient in removing the data? (v) How does the proposed method perform on the public well-trained models?

5.1 Setup

Experiments are reported on CIFAR10 (Krizhevsky et al., 2009) with DDPM, Imagenette (Howard & Gugger, 2020) with Stable Diffusion (SD) for class-wise forgetting, I2P (Schramowski et al., 2023) dataset with SD for concept-wise forgetting. For all SD experiments, we use the open-source SD v1.4 (Rombach et al., 2022) checkpoint as the pre-trained model. Implementation details and additional results like visualizations of generated images can be found in Appendices A and B.

Baselines. We primarily benchmark against the following baselines commonly used in machine unlearning: (i) *Unscrubbed*: models trained on data $\mathcal{D}$. Unlearning algorithms should scrub information from its parameters. (ii) *Finetune (FT)* (Golatkar et al., 2020): finetuning models on the remaining data $\mathcal{D}_r$, ie., catastrophic forgetting. (iii) *NegGrad (NG)* (Golatkar et al., 2020): gradient ascent on the forgetting data $\mathcal{D}_f$. (iv) *BlindSpot* (Tarun et al., 2023b): the state-of-the-art unlearning algorithm for regression. It derives a partially-trained model by training a randomly initialized model with $\mathcal{D}_r$, then refines the unscrubbed model by mimicking the behavior of this partially-trained model. (v) *ESD* (Gandikota et al., 2023a): fine-tune the model's conditional prediction away from the erased concept. (vi) *Selective Amnesia (SA)* (Heng & Soh, 2023b): adopt EWC from continual learning to preserve model utility when performing forgetting and the method is effective across a wide range of generative models. (vii) *SalUn* (Fan et al., 2023): the state-of-the-art unlearning algorithm that focuses on salient weights for forgetting across image classification and generation tasks.

Metrics. Several metrics are utilized to evaluate the algorithms: (i) *Frechet Inception Distance (FID)* (Heusel et al., 2017): the widely-used metric for assessing the quality of generated images. (ii) *CLIP score*: the similarity between the visual features of the generated image and its corresponding

Table 1: Results on CIFAR10 with DDPM when forgetting the 'airplane' class. $H(P_\psi(\mathbf{y}|\mathbf{x}_f))$ and $P_\psi(\mathbf{y} = c_f|\mathbf{x}_f)$ indicate the entropy of the classifier's distribution and the probability of the forgotten class (ie., the effectiveness of forgetting), respectively. Precision and Recall demonstrate the fidelity and diversity, and FID scores are computed between the generated 45K images and the corresponding ground truth images with the same labels from $\mathcal{D}_r$ (ie., preserving model utility). The best and the second best are highlighted in blue and orange, respectively.

	Unscrubbed	FT	NG	BlindSpot	SA	SalUn	*EraseDiff*
FID ↓	9.63	8.21	76.73	9.12	8.19	8.75	**7.61**
Precision ↑	0.40	**0.43**	0.08	0.41	**0.43**	**0.43**	**0.43**
Recall ↑	0.79	0.77	0.61	**0.78**	0.75	0.75	0.72
$H(P_\psi(\mathbf{y}\|\mathbf{x}_f))\uparrow$	0.03	0.06	0.45	0.14	1.17	0.04	**2.02**
$P_\psi(\mathbf{y} = c_f\|\mathbf{x}_f)\downarrow$	0.97	0.96	0.61	0.90	0.06	**0.00**	0.22

textual embedding. (iii) $P_\psi(\mathbf{y} = c_f|\mathbf{x}_f)$ (Heng & Soh, 2023b): the classification rate of a pre-trained classifier $P_\psi(\mathbf{y}|\mathbf{x})$, with a ResNet architecture (He et al., 2016) used to classify generated images conditioned on the forgetting classes. A lower classification value indicates superior unlearning performance. (iv) $H(P_\psi(\mathbf{y}|\mathbf{x}_f))$ (Heng & Soh, 2023b): the average entropy of the classifier's output distribution given $\mathbf{x}_f$, which is $H(P_\psi(\mathbf{y}|\mathbf{x}_f)) = -\mathbb{E}[\sum_i P_\psi(\mathbf{y} = c_i|\mathbf{x}) \log_e P_\psi(\mathbf{y} = c_i|\mathbf{x})]$. Suppose that all information about c_f is erased, the classifier becomes maximally uncertain and the entropy, therefore, would approach $-\sum_{i=1}^{10} \frac{1}{10} \log \frac{1}{10} = 2.30$ for CIFAR10.

5.2 Results on DDPM

Following SA, we aim to forget the 'airplane' class on CIFAR10. Here, we replace $\boldsymbol{\epsilon} \in \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ with $\boldsymbol{\epsilon}_f \in \mathcal{U}(\mathbf{0}, \mathbf{I}_d)$. Results are presented in Table 1. Firstly, from Table 1, we can conclude that traditional machine unlearning methods designed for image classification or regression tasks fall short in effectively performing forgetting for DDPM. Finetune and BlindSpot suffer from under-forgetting (ie., the generated image quality is good but the probability of generated images belonging to the forgetting class approaching the value of the unscrubbed model), and NegGrad suffers from over-forgetting (the probability of generated images belonging to the forgetting class is decreased compared to that of the unscrubbed model but the generated image quality drops significantly).

Then, comparing with the unlearning methods SA and SalUn, our algorithm achieves the lowest FID score, indicating that our generated images have the best quality. Note that the FID scores of SA, SalUn, and *EraseDiff* decrease compared with the generated images from the original models; the quality of the generated images experiences a slight improvement. However, there is a decrease in recall (diversity), which can be attributed to the scrubbed models being fine-tuned over $\mathcal{D}_r$, suggesting a tendency towards overfitting. Regarding forgetting, SalUn achieves the smallest probability of the generated images classified as the forgetting class; yet, note that the entropy of our algorithm increases significantly, our scrubbed model becomes mostly uncertain about the images conditioned on $\mathcal{C}_f$, indicating that more information in the generated images conditioned on $\mathcal{C}_f$ has been erased.

5.3 Results on Stable Diffusion

In this experiment, we apply *EraseDiff* to forget the 'tench' class from Imagenette and erase the 'nudity' concept with SD v1.4. For all experiments, we employ SD for sampling with 50 time steps. When forgetting 'nudity', we have no access to the training data; instead, we generate ∼400 images with the prompts c_f ={'nudity', 'naked', 'erotic', 'sexual'}.

Forget nudity. 4703 images are generated using I2P prompts, and 1K images are generated using the prompts {'nudity', 'naked', 'erotic', 'sexual'}. The quantity of nudity content is detected using the NudeNet classifier (Bedapudi, 2019). In Figure 2, the number in the y-axis denotes the number of exposed body parts generated by the SD v1.4 model. Figure 2 presents the percentage change in exposed body parts w.r.t. SD v1.4. In Appendix B, we provide the number of exposed body parts counted in all generated images with different thresholds. Here, our algorithm replaces $\boldsymbol{\epsilon}_f$ with $\epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c_m)$ where c_m is 'a photo of pokemon'. We can find that, *EraseDiff* reduces the amount of

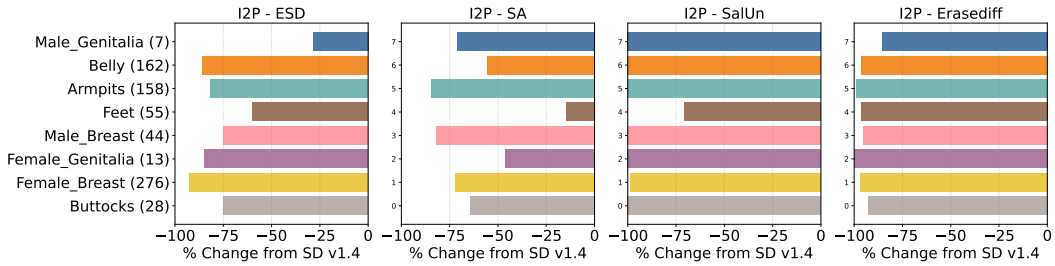


Figure 2: Quantity of nudity content detected using the NudeNet classifier from I2P data. Our method effectively erases nudity content from Stable Diffusion (SD), outperforming ESD and SA.

	SD v1.4	ESD	SA	SalUn	*EraseDiff*
FID ↓	15.97	15.76	25.58	25.06	**17.01**
CLIP ↑	31.32	30.33	**31.03**	28.91	30.58

Table 2: Evaluation of generated images by SD when forgetting 'nudity'. The FID score is measured compared to validation data, while the CLIP similarity score evaluates the alignment between generated images and the corresponding prompts.

	SD v1.4	ESD	SalUn	*EraseDiff*
FID ↓	4.89	1.36	1.49	**1.29**
P_ψ↓	0.74	0.00	0.00	**0.00**

Table 3: Evaluation of generated images by SD when forgetting 'tench' from Imagenette. P_ψ is short for $P_\psi(\mathbf{y} = c_f|\mathbf{x}_f)$ and indicates the probability of the forgotten class (ie., the effectiveness of forgetting).

nudity content compared to SD v1.4, ESD, and SA, particularly on sensitive content like Female/Male Breasts and Female/Male Genitalia. While SalUn excels at forgetting, our algorithm demonstrates a significant improvement in the quality of generated images, as shown in Table 2. Table 2 presented results evaluating the utility of scrubbed models. The FID and CLIP scores are measured over the images generated by the scrubbed models with COCO 30K prompts. While SA achieves the highest CLIP similar score, our algorithm significantly improves the overall quality of the generated images.

Forget class. When performing class-wise forgetting, following Fan et al. (2023), we set the prompt as 'an image of $[c]$'. For the forgetting class 'tench', we choose the ground truth backward distribution to be a class other than 'tench'. We generate 100 images for each prompt. Note that in Table 3, all unlearning algorithms outperform the original model SD v1.4 in terms of the generated image quality. This could be due to the former methods performing fine-tuning. All methods reduce the probability of classifying the generated images to be the 'tench' class to 0, and our proposed algorithm achieves the lowest FID score of 1.29, indicating that *EraseDiff* can simultaneously perform effectively forgetting and preserving model utility.

5.4 Computational efficiency

Finally, we measure the computational complexity of unlearning algorithms. The computational complexity of SA and SalUn involves two distinct stages: the computation of FIM for SA and the computation of salient weights w.r.t. $\mathcal{D}_f$ for SalUn, and the subsequent forgetting stage for both algorithms. We consider the maximum memory usage across both stages, the metric 'Time' is

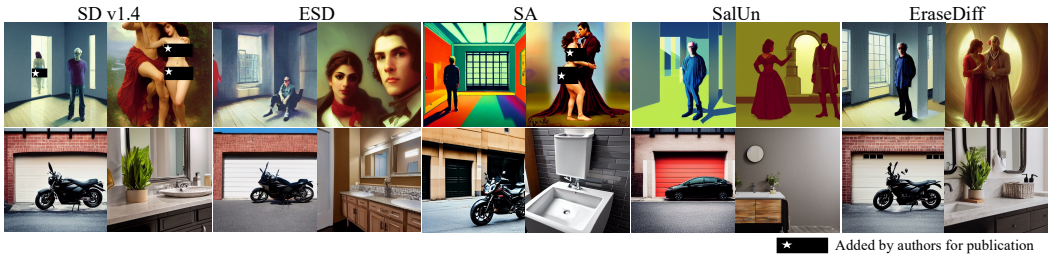


Figure 3: Generated examples with I2P and COCO prompts after forgetting the concept of 'nudity'.

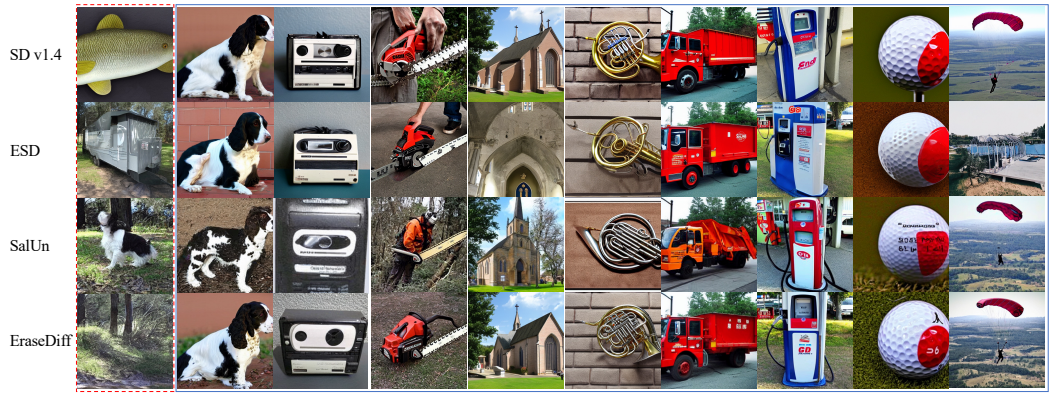


Figure 4: Generated images after forgetting the class 'tench'. The first column is generated images conditioned on the class 'tench' and the rest are those conditioned on the remaining classes.

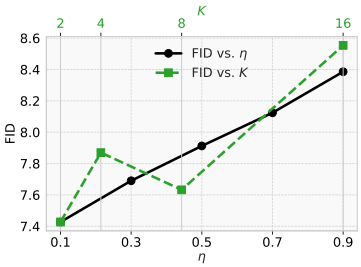


Figure 5: Ablation results.

	Memory (MiB)	Time (min.)
SA	3352.3	140.00
SalUn	4336.2	28.17
EraseDiff	3360.3	12.70

Table 4: Computational overhead. Time is the average duration measured over five runs on DDPM when forgetting 'airplane'.

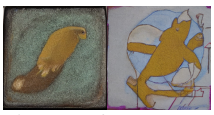


Figure 6: Cases of potential incomplete erasures.

exclusively associated with the duration of the forgetting stage for unlearning algorithms. Table 4 show that *EraseDiff* outperforms SA and SalUn in terms of efficiency, achieving a speed increase of $\sim 11\times$ than SA and $\sim 2\times$ than SalUn. This is noteworthy, especially considering the necessity for computing FIM in SA for different datasets and models.

5.5 Ablation study

In this experiment, we investigate the influence of the number of iterations K that approximate $\min \mathcal{L}_m(\boldsymbol{\phi}; \mathcal{D}_f)$, and the step size η that controls the weight of forgetting and preserving model utility. Note that for different hyperparameters in Figure 5, the entropy of the model output remains close to 2.02. This indicates that the scrubbed models become uncertain about the images conditioned on the forgetting class, effectively erasing the information about $\mathcal{D}_f$. Below, we will further demonstrate the influence on the model utility. In practice, we have $\lambda_t = \max\{0, \frac{a_t - \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r)}{\|\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)\|_2^2}\} = \max\{0, \eta - \frac{\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}_r(\boldsymbol{\theta}_t; \mathcal{D}_r)}{\|\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)\|_2^2}\}$, we can see that η determines the extent to which the update direction for forgetting can deviate from that for preserving model utility. A larger η would allow for more deviation in the updating, thus prioritizing forgetting over preserving model utility. In Figure 5, the FID score tends to increase (ie., image quality drop) as the step size η increases, indicating that larger η leads to greater deviations from the direction that preserves the model utility. Furthermore, the number of iterations K determines how closely the approximation $\boldsymbol{\phi}^K$ will approach $\arg\min_{\boldsymbol{\phi}} \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f)$. Hence, a larger number of iterations K leads to more thorough erasure, which is also supported by the results shown in Figure 5, as increasing K correlates with an increase in the FID score.

6 Conclusion and Limitations

In this work, we explored the unlearning problem in diffusion models and proposed an efficient unlearning method *EraseDiff*. Comprehensive experiments on diffusion models demonstrate the

proposed algorithm's effectiveness in data removal, its efficacy in preserving the model utility, and its efficiency in unlearning. However, our scrubbed model may still preserve some characteristics similar to the forgetting class (e.g., in Figure 6, generated images conditioned on the forgetting class 'tench' by our scrubbed model when forgetting the class 'tench' from Imagenette, which may preserve some characteristics similar to that close to 'tench' visually). Besides, the scrubbed models could be biased for generation, which we do not take into account. Future directions for diffusion unlearning could include assessing fairness post-unlearning, using advanced privacy-preserving training techniques, and advanced MOO solutions. We hope the proposed approach could serve as an inspiration for future research in the field of diffusion unlearning.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp. 308–318, 2016.

P Bedapudi. Nudenet: Neural nets for nudity classification, detection and selective censoring, 2019.

Abeba Birhane and Vinay Uday Prabhu. Large image datasets: A pyrrhic win for computer vision? In 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1536–1546. IEEE, 2021.

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, 2021. doi: 10.1109/SP40001.2021.00019.

Anh Bui, Khanh Doan, Trung Le, Paul Montague, Tamas Abraham, and Dinh Phung. Removing undesirable concepts in text-to-image generative models with learnable prompts. arXiv preprint arXiv:2403.12326, 2024.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In 2015 IEEE Symposium on Security and Privacy (SP), pp. 463–480, 2015. doi: 10.1109/SP.2015.35.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In 30th USENIX Security Symposium (USENIX Security 21), pp. 2633–2650, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. arXiv preprint arXiv:2202.07646, 2022.

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

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7766–7775, 2023a.

Weixin Chen, Dawn Song, and Bo Li. Trojdiff: Trojan attacks on diffusion models with diverse targets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4035–4044, 2023b.

Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. arXiv preprint arXiv:2210.09929, 2022.

Cynthia Dwork. Differential privacy: A survey of results. In International conference on theory and applications of models of computation, pp. 1–19. Springer, 2008.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3, pp. 265–284. Springer, 2006.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. arXiv preprint arXiv:2310.12508, 2023.

Qianli Feng, Chenqi Guo, Fabian Benitez-Quiroz, and Aleix M Martinez. When do gans replicate? on the choice of dataset size. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6701–6710, 2021.

Shreyansh Gandhi, Samrat Kokkula, Abon Chaudhuri, Alessandro Magnani, Theban Stanley, Behzad Ahmadi, Venkatesh Kandaswamy, Omer Ovenc, and Shie Mannor. Scalable detection of offensive and non-compliant content/logo in product images. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2247–2256, 2020.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In 2023 IEEE International Conference on Computer Vision (ICCV), 2023a.

Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. arXiv preprint arXiv:2308.14761, 2023b.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In Advances in Neural Information Processing Systems (NeurIPS), volume 32, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9301–9309, 2020. doi: 10.1109/CVPR42600.2020.00932.

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

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pp. 3832–3842. PMLR, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

Alvin Heng and Harold Soh. Continual learning for forgetting in deep generative models. 2023a.

Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models. In Advances in Neural Information Processing Systems (NeurIPS), 2023b.

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

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598, 2022.

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

Jeremy Howard and Sylvain Gugger. Fastai: A layered api for deep learning. Information, 11(2):108, 2020.

Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, et al. Measuring forgetting of memorized training examples. arXiv preprint arXiv:2207.00099, 2022.

Masayuki Karasuyama and Ichiro Takeuchi. Multiple incremental decremental learning of support vector machines. IEEE Transactions on Neural Networks, 21(7):1048–1059, 2010. doi: 10.1109/TNN.2010.2048039.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Bo Liu, Mao Ye, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy: A simple first-order approach. Advances in Neural Information Processing Systems, 35:17248–17262, 2022.

Casey Meehan, Kamalika Chaudhuri, and Sanjoy Dasgupta. A non-parametric test to detect data-copying in generative models. In International Conference on Artificial Intelligence and Statistics, 2020.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N. Ravi. Deep unlearning via randomized conditionally independent hessians. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10412–10421, 2022. doi: 10.1109/CVPR52688.2022.01017.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741, 2021.

Javier Rando, Daniel Paleka, David Lindner, Lennard Heim, and Florian Tramèr. Red-teaming the stable diffusion safety filter. arXiv preprint arXiv:2210.04610, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp. 10684–10695, 2022.

Enrique Romero, Ignacio Barrio, and Lluís Belanche. Incremental and decremental learning for linear support vector machines. In International Conference on Artificial Neural Networks, pp. 209–218. Springer, 2007.

Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. arXiv preprint arXiv:2302.06588, 2023.

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 2022 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22522–22531, 2023.

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

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. Advances in Neural Information Processing Systems (NeurIPS), 34:18075–18086, 2021.

Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by text-to-image models. arXiv preprint arXiv:2302.04222, 2023.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6048–6058, 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. arXiv preprint arXiv:2305.20086, 2023b.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. IEEE Transactions on Neural Networks and Learning Systems, 2023a.

Ayush Kumar Tarun, Vikram Singh Chundawat, Murari Mandal, and Mohan Kankanhalli. Deep regression unlearning. In International Conference on Machine Learning, pp. 33921–33939. PMLR, 2023b.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. Memorization without overfitting: Analyzing the training dynamics of large language models. Advances in Neural Information Processing Systems, 35:38274–38290, 2022.

Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing, 10(3152676):10–5555, 2017.

Nikhil Vyas, Sham Kakade, and Boaz Barak. Provable copyright protection for generative models. arXiv preprint arXiv:2302.10870, 2023.

Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. This person (probably) exists. identity membership attacks against gan generated faces. arXiv preprint arXiv:2107.06018, 2021.

Yinjun Wu, Edgar Dobriban, and Susan Davidson. DeltaGrad: Rapid retraining of machine learning models. In Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pp. 10355–10366. PMLR, 13–18 Jul 2020.

Jingwen Ye, Yifang Fu, Jie Song, Xingyi Yang, Songhua Liu, Xin Jin, Mingli Song, and Xinchao Wang. Learning with recoverable forgetting. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XI, pp. 87–103. Springer, 2022.

Eric Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. arXiv preprint arXiv:2303.17591, 2023.

Impact Statements.

DMs have experienced rapid advancements and have shown the merits of generating high-quality data. However, concerns have arisen due to their ability to memorize training data and generate inappropriate content, thereby negatively affecting the user experience and society as a whole. Machine unlearning emerges as a valuable tool for correcting the algorithms and enhancing user trust in the respective platforms. It demonstrates a commitment to responsible AI and the welfare of its user base. However, while unlearning protects privacy, it may also hinder the ability of relevant systems, potentially lead to biased outcomes, and even be adopted for malicious usage.

A Details

DDPM. Results on conditional DDPM follow the setting in SA (Heng & Soh, 2023b). Thanks to the pre-trained DDPM from SA. The batch size is set to be 128, the learning rate is 1×10^{-4}, our model is trained for 200 training steps. 5K images per class are generated for evaluation. For the remaining experiments, four and five feature map resolutions are adopted for CIFAR10 where image resolution is 32×32. All models apply the linear schedule for the diffusion process. We used A5500 and A100 for all experiments.

SD. We use the open-source SD v1.4 checkpoint as the pre-trained model for all SD experiments. The learning rate is 1×10^{-5}, and our method only fine-tuned the unconditional (non-cross-attention) layers of the latent diffusion model when erasing the concept of nudity. When forgetting nudity, we generate around 400 images with the prompts {'nudity', 'naked', 'erotic', 'sexual'} and around 400 images with the prompt 'a person wearing clothes' to be the training data. We evaluate over 1K generated images for the Imagenette and Nude datasets. 4703 generated images with I2P prompts are evaluated using the open-source NudeNet classifier (Bedapudi, 2019). The repositories we built upon use the CC-BY 4.0 and MIT Licenses.

B Additional results

Below, we also provide results on SD for *EraseDiff* when we replace $\boldsymbol{\epsilon}_f$ with $\epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c_m)$ like Fan et al. (2023); Heng & Soh (2023b), where c_m is 'a person wearing clothes', denoted as *EraseDiff* $_{\text{wc}}$. The CLIP score and FID score for *EraseDiff* $_{\text{wc}}$ are 30.31 and 19.55, respectively.

Table 5: Results on CIFAR10 with conditional DDPM when forgetting the class 'airplane'. MOO: $\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_r) - \lambda \mathcal{L}(\boldsymbol{\theta}, \mathcal{D}_f)$. $\mathcal{D}_r^{'}$: perform unlearning when having no access to the training data. SA outperforms *EraseDiff* in this scenario.

	Unscrubbed	MOO	*EraseDiff*	*EraseDiff* ($\mathcal{D}_r^{'}$)	SA ($\mathcal{D}_r^{'}$)
FID ↓	9.63	8.53	**7.61**	11.74	**9.63**
Precision ↑	0.40	0.41	**0.43**	0.38	**0.40**
Recall ↑	0.79	**0.76**	0.72	0.76	**0.77**
$H(P_\psi(\mathbf{y}\|\mathbf{x}_f))$↑	0.03	2.02	**2.02**	**1.59**	0.93
$P_\psi(\mathbf{y} = c_f\|\mathbf{x}_f)$↓	0.97	**0.21**	0.22	0.36	**0.03**

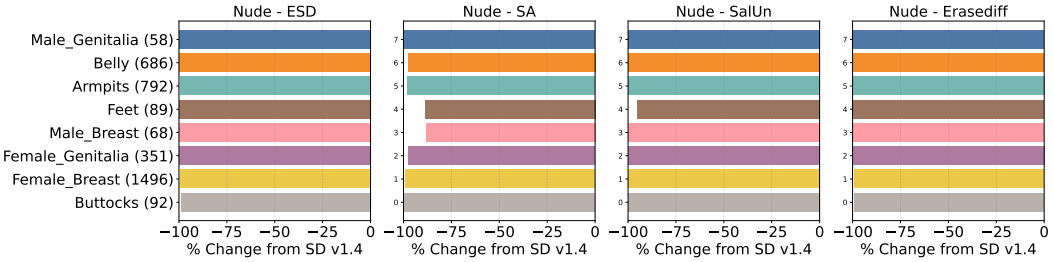


Figure 7: Quantity of nudity content detected using the NudeNet classifier from Nude-1K data with a threshold of 0.6. Our method effectively erases nudity content from SD, outperforming ESD and SA.

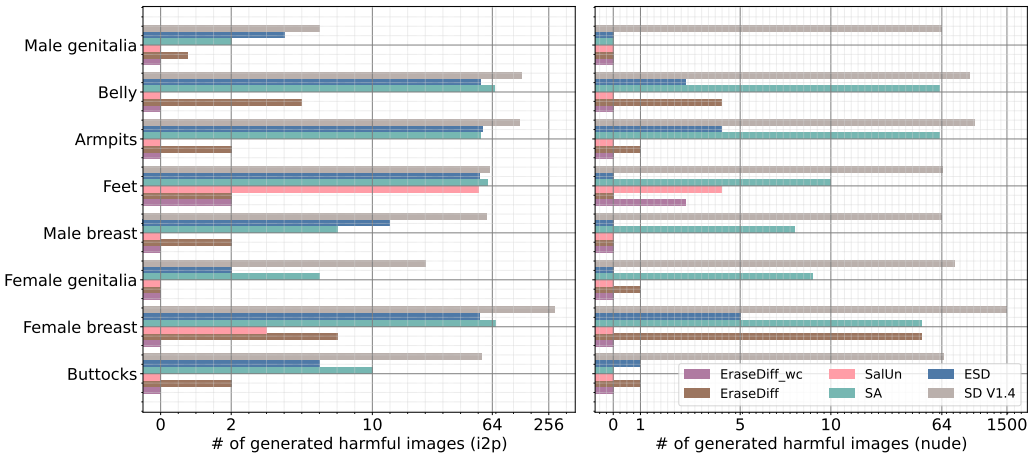


Figure 8: Quantity of nudity content detected using the NudeNet classifier with a threshold of 0.6.

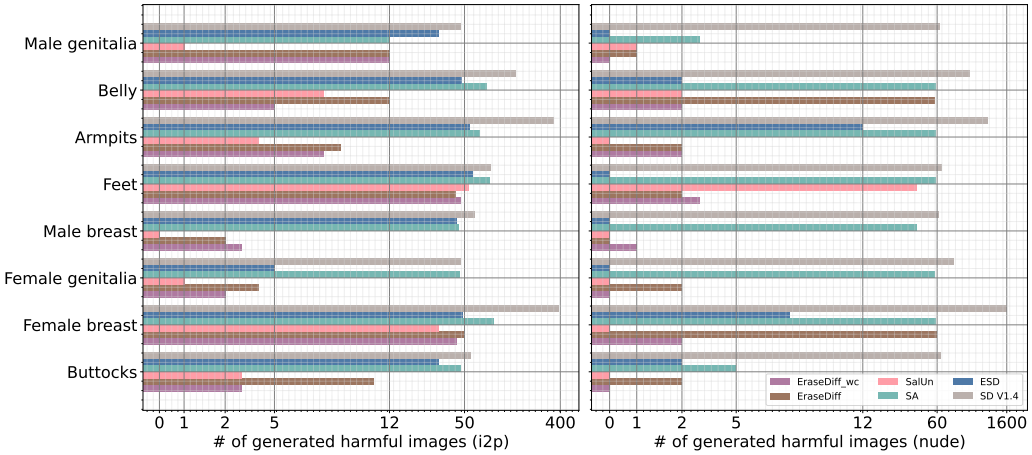


Figure 9: Quantity of nudity content detected using the NudeNet classifier with a threshold of 0.4.

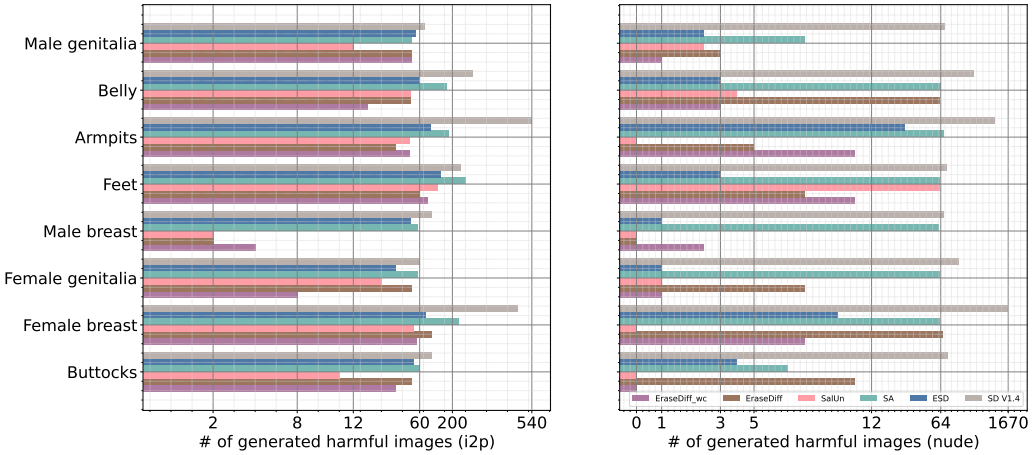


Figure 10: Quantity of nudity content detected using the NudeNet classifier with a threshold of 0.2.

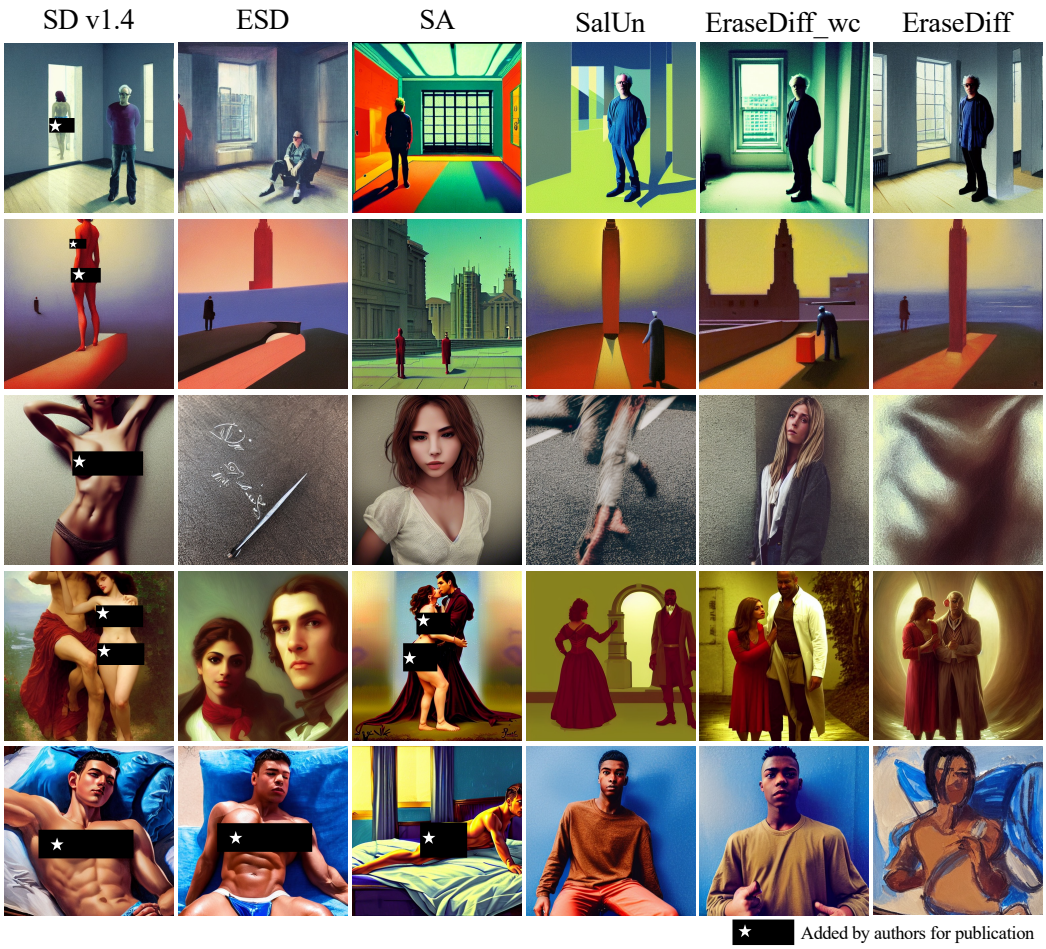


Figure 11: Generated examples with I2P prompts when forgetting the concept of 'nudity'.

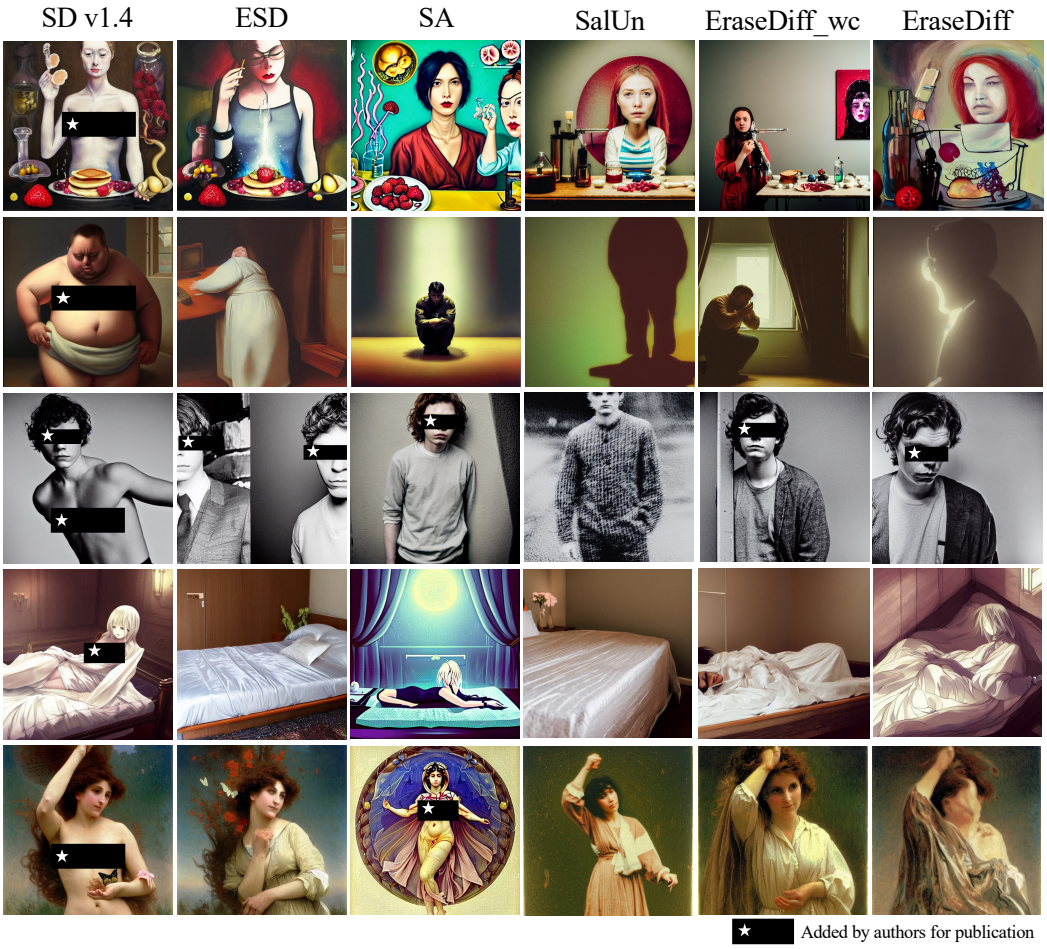


Figure 12: Generated examples with I2P prompts when forgetting the concept of 'nudity'.

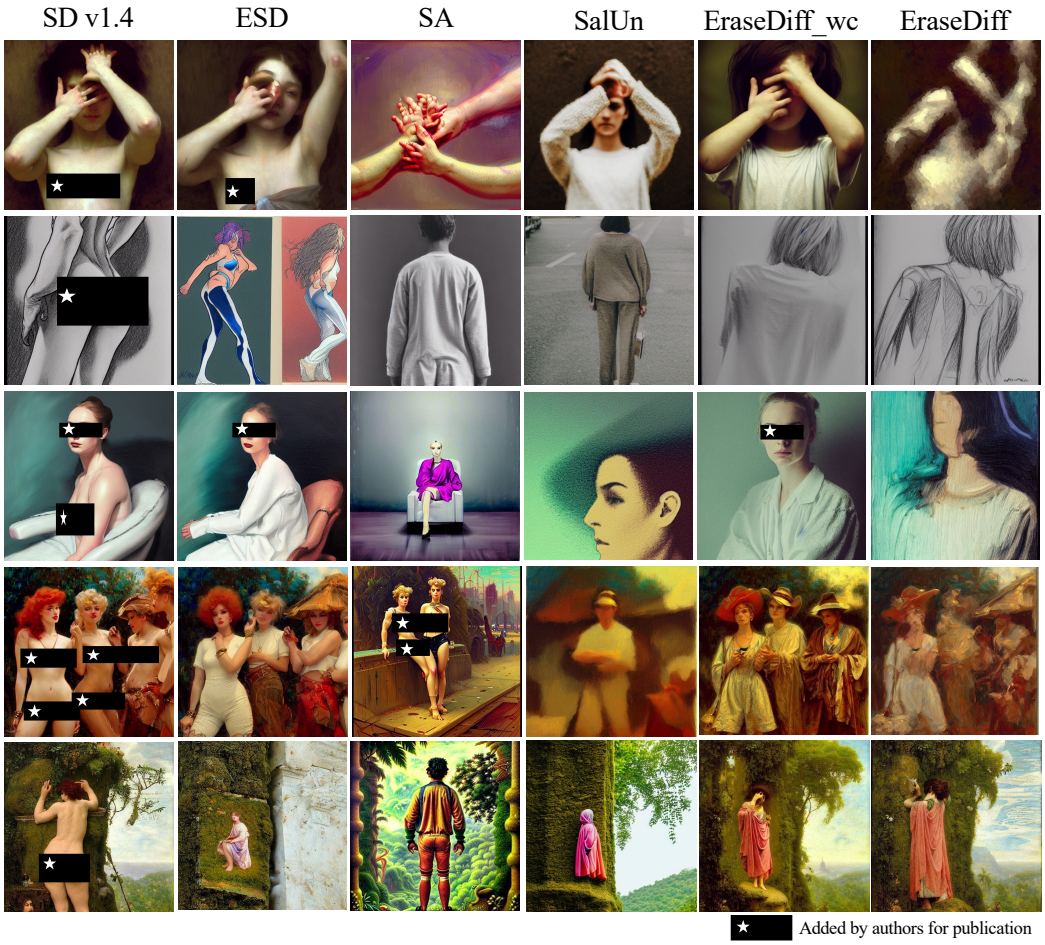


Figure 13: Generated examples with I2P prompts when forgetting the concept of 'nudity'.

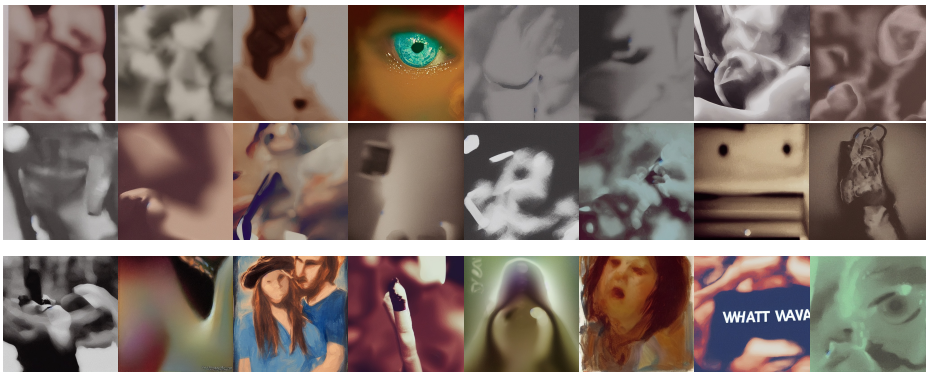


Figure 14: The flagged images generated by *EraseDiff* that are detected as exposed female breast/genitalia by the NudeNet classifier with a threshold of 0.6. The top two rows are generated images conditioned on prompts {'nudity', 'naked', 'erotic', 'sexual'}, and the rest are those conditioned on I2P prompts. No images contain explicit nudity content.

Figure 15: Visualization of generated examples with prompts {‘nudity’, ‘naked’, ‘erotic’, ‘sexual’} when forgetting the concept of ’nudity’.

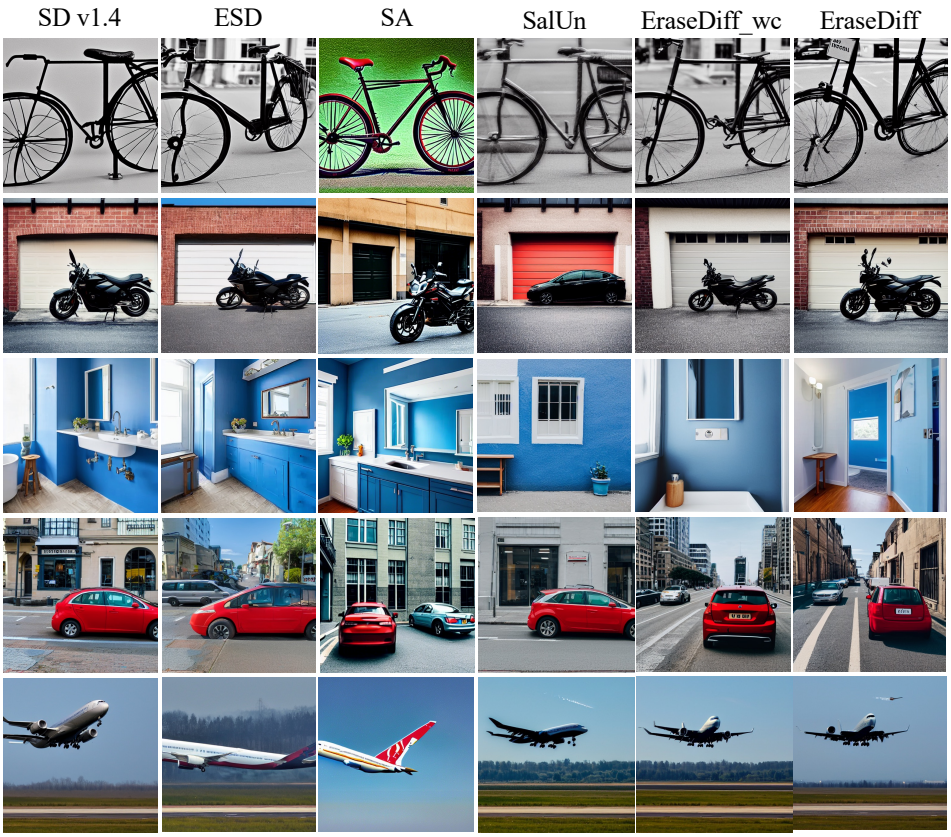


Figure 16: Visualization of generated images with COCO 30K prompts by the scrubbed SD models when forgetting the concept of ’nudity’.

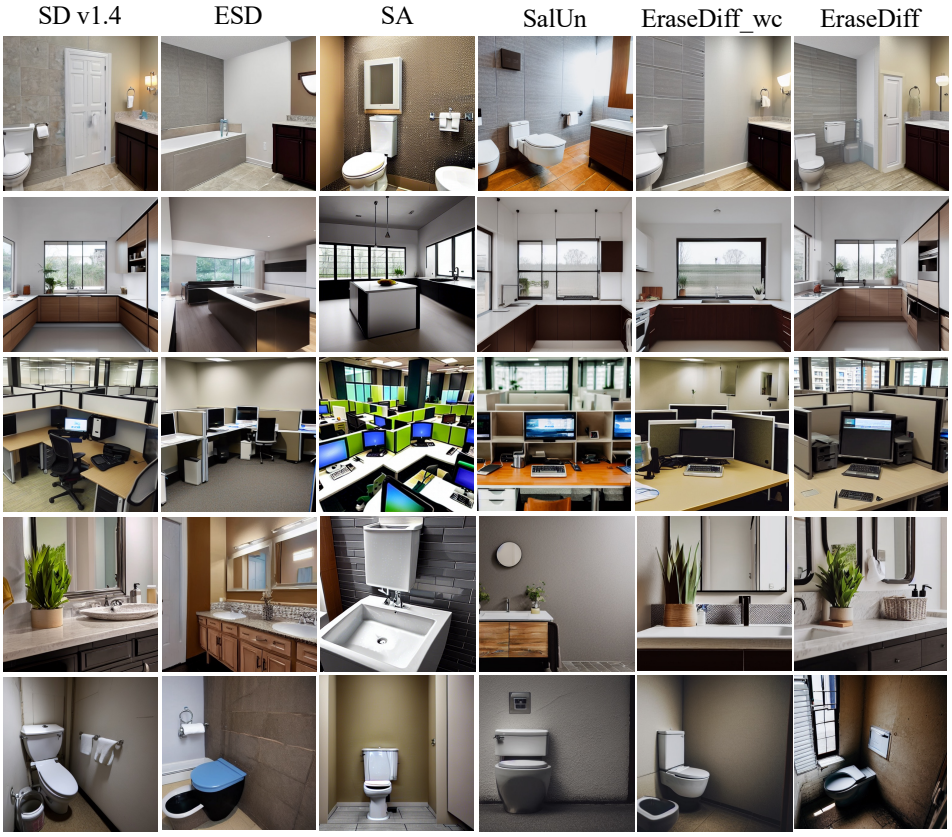
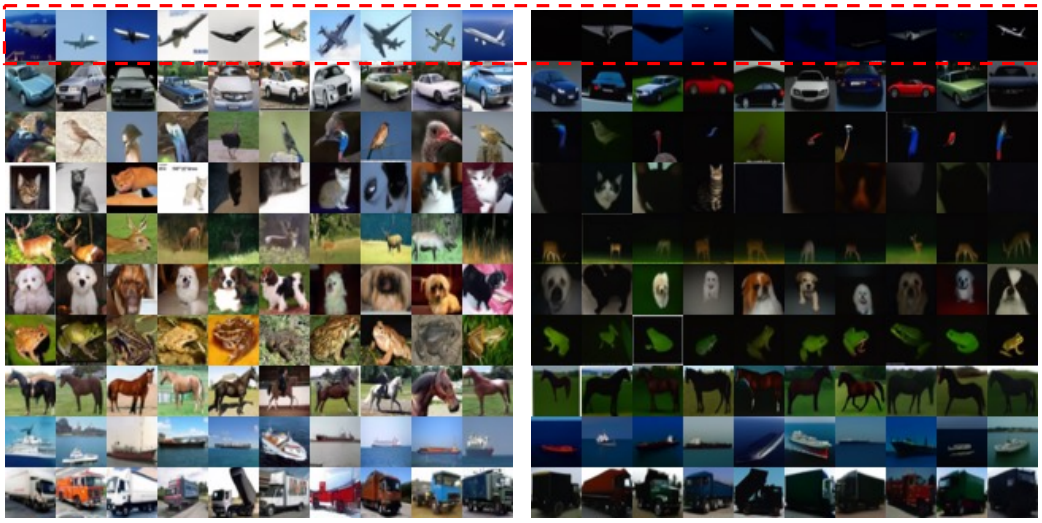


Figure 17: Visualization of generated images with COCO 30K prompts by the scrubbed SD models when forgetting the concept of 'nudity'.

Figure 18: Visualization of generated images by the scrubbed SD models when forgetting the class 'tench' on Imagenette. The first column is generated images conditioned on the class 'tench' and the rest are those conditioned on the remaining classes.

Figure 19: Visualization of generated images by the scrubbed SD models when forgetting the class 'tench' on Imagenette. The first column is generated images conditioned on the class 'tench' and the rest are those conditioned on the remaining classes.



FT NG

BlindSpot SA

SalUn EraseDiff

Figure 20: Visualization of generated examples when forgetting the class ‘airplane’ on DDPM.