Erasing Undesirable Influence in Diffusion Models

Jing Wu[1], Trung Le[1], Munawar Hayat[2], Mehrtash Harandi[1]
[1]Monash University, Melbourne, VIC, Australia, [2]Qualcomm, San Deigo, CA, US
{jing.wu1, trunglm, mehrtash.harandi}@monash.edu, hayat@qti.qualcomm.com

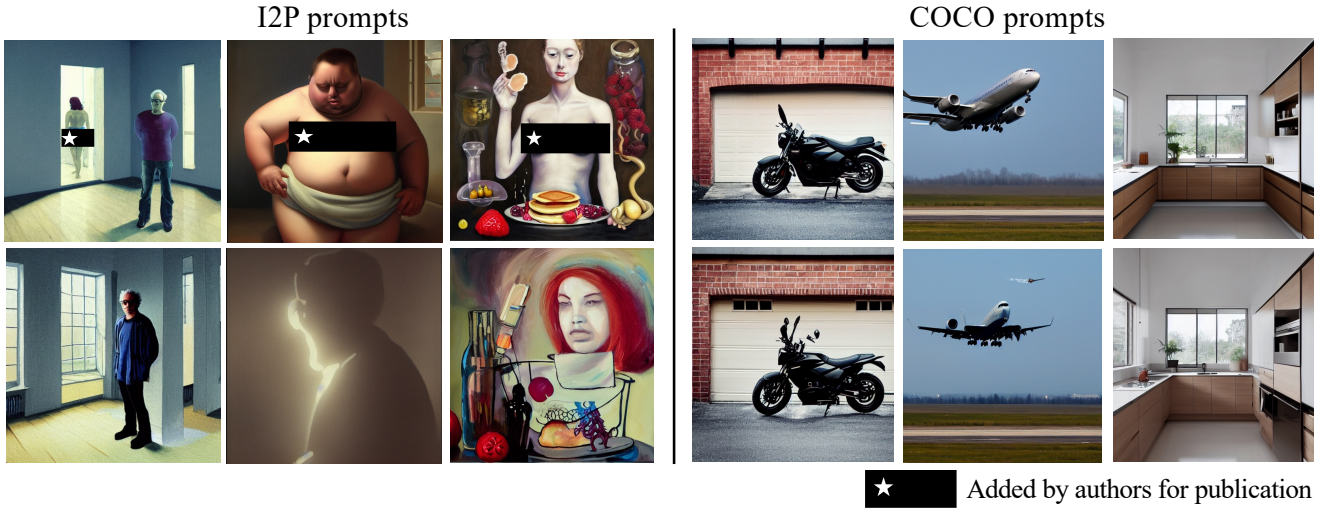

Figure 1. Top to Bottom: generated samples by SD v1.4 and model scrubbed by our method, when erasing the concept of 'nudity'. Our method can avoid NSFW (not safe for work) content while preserving model utility.

Abstract

Diffusion models are highly effective at generating high-quality images but pose risks, such as the unintentional generation of NSFW (not safe for work) content. Although various techniques have been proposed to mitigate unwanted influences in diffusion models while preserving overall performance, achieving a balance between these goals remains challenging. In this work, we introduce EraseDiff, an algorithm designed to preserve the utility of the diffusion model on retained data while removing the unwanted information associated with the data to be forgotten. Our approach formulates this task as a constrained optimization problem using the value function, resulting in a natural first-order algorithm for solving the optimization problem. By altering the generative process to deviate away from the ground-truth denoising trajectory, we update parameters for preservation while controlling constraint reduction to ensure effective erasure, striking an optimal trade-off. 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 [30, 49, 60] are now the method of choice in deep generative models, owing to their high-quality output, stability, and ease of training 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 [10, 58, 59]. Beyond privacy, diffusion models are susceptible to misuse and can generate NSFW digital content [48, 52, 54]. 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) [64] and the California Consumer Privacy Act (CCPA) [24] 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 [4, 11, 23, 25, 44, 56, 61, 62, 69].

A straightforward solution 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 [55] costs approximately 150,000 GPU hours with 256 A100 GPUs[1]. Existing research on erasing unwanted influence has primarily focused on classification tasks [4, 7, 11, 22, 23, 25, 33, 44, 56, 68]. Despite substantial progress, prior methods developed in classification are observed to be ineffective for generation tasks [16]. 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 [6, 16, 20, 21, 27, 28, 36, 41, 71] target unlearning in diffusion models, with a primary focus on the text-to-image models [20, 21, 71]. Broadly, these methods aim to achieve two main objectives: erasing data influence and preserving overall model performance. However, as demonstrated by Bui et al. [5], balancing this trade-off remains challenging.

In this work, we propose *EraseDiff*, an algorithm tailored to balance the overall performance of diffusion models with the erasure of undesirable information. Drawing inspiration from optimization-based meta-learning algorithms [18, 42] that enable fast adaptation to new learning tasks, we formulate this challenge as a bi-level optimization problem, where the "inner" optimization focus on erasing undesirable influence and the "outer" objective seeks to preserve model performance. The outer objective and inner optimization are interdependent, iterating between preservation and erasure to balance the trade-off effectively. However, this nested optimization can be challenging to optimize efficiently. The inner optimization may converge to a saddle point or struggle with non-convex functions, making it difficult to achieve a stable solution [39]. Therefore, we further reformulate the problem as a constrained optimization problem using the value function [40, 46, 70], which facilitates a natural first-order solution [39], allows us to optimize preservation and erasure in a unified manner. This approach achieves a fine-tuned balance between preservation and targeted erasure, yielding an optimal trade-off. We benchmark *EraseDiff* on various scenarios, encompassing unlearning of classes on CIFAR-10 [35] with Denoising Diffusion Probabilistic Models (DDPM) [30], classes on Imagenette [31] and concepts on the I2P dataset [54] with stable diffusion. Our empirical findings show that *EraseDiff* is 11× faster than Heng and Soh's method [28] and 2× faster than Fan's method [16] 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 and latent diffusion models [49]. 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, i.e., $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}_d)$, 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, i.e., $\|\boldsymbol{\epsilon} - \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\|$. (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.

Latent diffusion models. Latent diffusion models 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, text embeddings generated by models like CLIP are used as 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:

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

where $\boldsymbol{\epsilon}_f = \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t | c_m)$ and $c_m \neq c$ so that the denoised image $\mathbf{x}_0$ is not related to the forgetting class/concept c [16, 28]. With this, we hinder the approximator $\epsilon_{\boldsymbol{\theta}}$ to guide the denoising process to obtain meaningful examples for the forgetting data example $\mathbf{x}_0 \sim \mathcal{D}_f$.

To erase the undesirable influence of $\mathcal{D}_f$ and preserve the overall performance, 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 $\lambda > 0$ as the optimization objective (see for example [16]). However, training could be hindered due to the conflicting gradients between the erasing and preservation objectives, preventing a balanced trade-off between erasure and preservation [38]. To address this, rather than scalarizing the two objectives, we consider a framework based on optimization-based meta-learning algorithm [47] that allows iteratively updates to optimize each objective:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \ \mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r) \\ &\text{s.t.} \quad \boldsymbol{\theta} \in \arg\min_{\boldsymbol{\phi}} \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f) \ , \end{aligned} \tag{4}$$

where the outer objective minimizes the remaining loss $\mathcal{L}_r$ (i.e., preserving model utility), the inner optimization minimizes the forgetting loss $\mathcal{L}_f$ (i.e., erasing) with initialization $\mathbb{R}^n \ni \boldsymbol{\phi}_{\texttt{init}} = \boldsymbol{\theta}$. Given $\boldsymbol{\theta}$, the inner optimization on $\boldsymbol{\phi}$ aims to minimize the forgetting data influence, with the goal of achieving effective erasure while preserving model utility. The outer objective and inner optimization are interdependent, iterating between preservation and erasure to balance the trade-off effectively.

While the above framework allows for iterative updates to address the conflicting objectives of erasure and preservation, it still relies on nested optimization, which can be challenging to optimize efficiently. The inner optimization may converge to a saddle point or struggle with non-convex functions, making it difficult to achieve a stable solution [39]. To further streamline the optimization process, we adopt a value function approach [40, 46, 70] that reformulates the problem as a single-constrained optimization:

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

where $\boldsymbol{\phi}$ is initialized at $\boldsymbol{\theta}$, leverages the value function to encapsulate the influence of data erasure directly as a constraint on $\mathcal{L}_f$. This avoids the need for a nested loop by capturing the forgetting objective as a constraint and provides a natural first-order solution, as the constrained formulation allows us to optimize preservation and erasure in a unified manner.

3.2. Solution

To solve Eq. (5), let us first denote $g(\boldsymbol{\theta}) := \mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f) - \min_{\boldsymbol{\phi}} \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f)$. Our goal is to erase undesirable influence while preserving the overall model performance, hence the update vector $\boldsymbol{\delta}_t$ for updating the model should aid in minimizing $\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r)$ and $g(\boldsymbol{\theta})$ simultaneously. In other words, suppose that the current solution for Eq. (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 (i.e., preserve model utility) and $g(\boldsymbol{\theta}_{t+1})$ decreases (i.e., erasure). To this end, we aim to find the update vector $\boldsymbol{\delta}_t$ 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 $g(\boldsymbol{\theta}_{t+1}) - g(\boldsymbol{\theta}_t) \approx -\eta \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)^\top \boldsymbol{\delta} \leq -\eta a_t < 0$ for some scalar $a_t > 0$, we can ensure that $g(\boldsymbol{\theta}_{t+1}) < g(\boldsymbol{\theta}_t)$ 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 Eq. (6), the following theorem is developed:

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

We provide the proof in §7 in the Appendix. This provides the solution to the optimization problem by constructing the update vector $\boldsymbol{\delta}$ to balance two competing objectives. The variable a_t adjusts the weight of the forgetting objective, ensuring that the update vector $\boldsymbol{\delta}$ decreases the remaining loss without violating the erasure goal, and achieves the dual goals of maintaining utility and achieving erasing. In practice, we can choose $a_t = \eta \|\nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_t)\|_2^2$, and 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 $k = 0, \cdots, K-1$.

Algorithm 1 *EraseDiff*: Erasing undesirable 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: $\boldsymbol{\phi}^0 = \boldsymbol{\theta}_t$.
3: 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: Set $g(\boldsymbol{\theta}_t) = \mathcal{L}_f(\boldsymbol{\theta}_t; \mathcal{D}_f) - \mathcal{L}_f(\boldsymbol{\phi}^K; \mathcal{D}_f)$.
5: 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: 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**

3.3. Analysis

We can characterize the solution of our algorithm as follows and the proof can be found in §7 in the Appendix:

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

This asserts that the solution obtained by our algorithm is Pareto optimal for the problem of minimizing both objectives, which implies that the solution obtained by the algorithm ensures a balanced trade-off between preserving model utility and erasing undesirable influences.

We further take DDPM with CIFAR-10 when forgetting the 'airplane' as an example to show that our proposed method helps alleviate the gradient conflict which prevents a balanced trade-off between erasure and preservation. Fig. 2 presents the cosine similarity between the update vector $\boldsymbol{\delta}$ and the gradient $\boldsymbol{g}_r = \nabla_{\boldsymbol{\theta}}\mathcal{L}_r(\boldsymbol{\theta}; \mathcal{D}_r)$ for preservation, and the cosine similarity between the update vector $\boldsymbol{\delta}$ and the gradient $\boldsymbol{g}_f = \nabla\mathcal{L}_f(\boldsymbol{\theta}; \mathcal{D}_f)$ for erasing. The cosine similarity represents the alignment between the update vector and the gradients associated with the preservation and erasure. Higher positive values indicate alignment, meaning that the update vector $\boldsymbol{\delta}$ is directed similarly to the respective gradient, whereas negative values indicate misalignment or conflict. In particular, negative values suggest a high degree of opposition between the update vector and the respective gradient, which can signify competing objectives between preservation and erasure during the optimization process.

MOO (Multi-Objective Optimization) denotes the naive integration of erasing and preservation as stated in Eq. (3). For the vanilla MOO, Fig. 2 shows a clear alternating pattern in cosine similarity values between the update vector $\boldsymbol{\delta}$ and the gradients $\boldsymbol{g}_r$ and $\boldsymbol{g}_f$. Specifically, when the cosine similarity between $\boldsymbol{\delta}$ and $\boldsymbol{g}_r$ is greater than 0, the similarity between $\boldsymbol{\delta}$ and $\boldsymbol{g}_f$ tends to be less than 0, and vice versa. This pattern suggests that MOO experiences gradient conflict, as it cannot effectively balance the two objectives of preservation and erasure, preventing MOO from achieving a harmonious update that supports both goals simultaneously. In contrast, *EraseDiff* mostly shows positive cosine similarity values between the update vector $\boldsymbol{\delta}$ and both the preservation gradient $\boldsymbol{g}_r$ and the erasing gradient $\boldsymbol{g}_f$. This indicates that *EraseDiff* aligns the update direction with both objectives, suggesting it manages to avoid significant gradient conflict. By maintaining positive alignment, *EraseDiff* appears to balance preservation and erasure more effectively, leading to better cooperation between objectives.

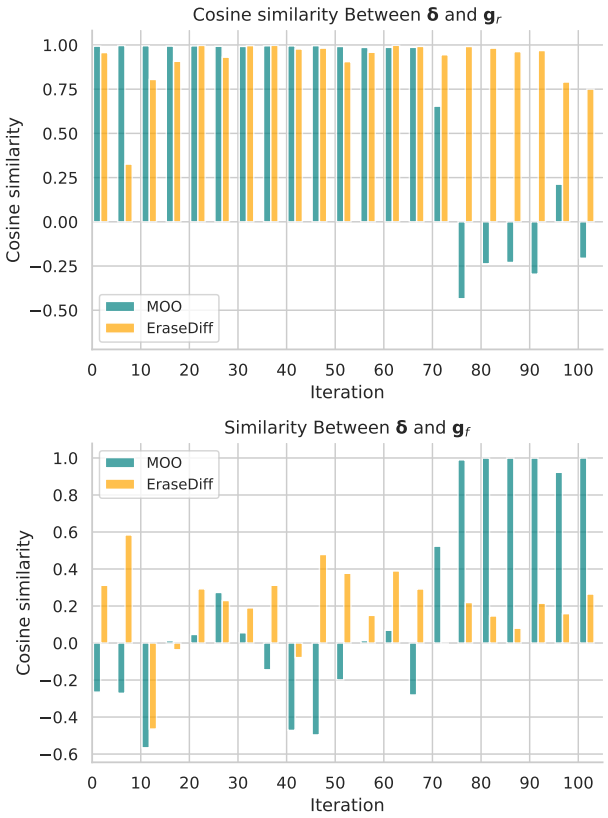

Figure 2. Top to bottom: cosine similarity between the update vector $\boldsymbol{\delta}$ and the preservation gradient $\boldsymbol{g}_r$, followed by the cosine similarity between $\boldsymbol{\delta}$ and the erasing gradient $\boldsymbol{g}_f$. Positive values indicate alignment, while negative values suggest conflict. This visualization illustrates how well the update vector aligns with the objectives of preservation and erasure over successive iterations.

4. Related Work

Memorization in generative models. Privacy of generative models has been studied extensively for GANs [17, 43,

66] and generative language models [8, 9, 32, 63]. These generative models often risk replicating from their training data. Recently, several studies [10, 58, 59, 65] 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 [58, 59], or training models with differential privacy (DP) [1, 13–15]. However, leveraging DP-SGD [1] may cause training to diverge [10].

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 [12] or manipulated [48], resulting in inappropriate generation [54]. They also risk the imitation of copyrighted content, e.g., mimicking the artistic style [20, 57]. To counter inappropriate generation, data censoring [3, 19, 45, 53] where excluding black-listed images before training, and safety guidance where diffusion models will be updated away from the inappropriate/undesired concept [20, 54] are proposed. [34] uses human feedback to annotate the generations for getting soft tokens that quantify the harmfulness of the images, then fine-tunes the model using self-distillation to achieve harmony between preservation and erasure. Shan et al. [57] propose protecting artistic style by adding barely perceptible perturbations to the artworks before public release. Yet, Rando et al. [48] argue that DMs can still generate content that bypasses the filter. Chen et al. [12] 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 [4, 7, 11, 22, 23, 25, 33, 44, 50, 56, 61, 68] have been proposed. Several studies [5, 16, 20, 21, 27, 28, 67, 71] recently introduce unlearning in diffusion models. Most of them [20, 21, 27, 71] mainly focus on text-to-image models and high-level visual concept erasure. Heng and Soh [28] 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. [16] 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, to achieve a fine-tuned balance between preservation and targeted erasure, yielding an optimal trade-off. Below, we will show that our algorithm is not only faster than Heng and Soh's method [28] and Fan's method [16], 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 *EraseDiff* able to remove the influence of $\mathcal{D}_f$ in the diffusion models? (iii) Is *EraseDiff* able to preserve the model utility while removing $\mathcal{D}_f$? (iv) Is *EraseDiff* efficient in removing the data? (v) How does *EraseDiff* perform on the public well-trained models?

5.1. Setup

Experiments are reported on CIFAR-10 [35] with DDPM, Imagenette [31] with Stable Diffusion (SD) for class-wise forgetting, I2P [54] dataset with SD for concept-wise forgetting. For all SD experiments, we use the open-source SD v1.4 [49] checkpoint as the pre-trained model. Implementation details and additional results like visualizations of generated images can be found in §8 and §9.

Baselines. We primarily benchmark against the following baselines commonly used in machine unlearning: (i) *Unscrubbed*, (ii) *Finetune (FT)* [23], (iii) *NegGrad (NG)* [23], (iv) *BlindSpot* [61], (v) *ESD* [20], (vi) *FMN* [71], (vii) *Selective Amnesia (SA)* [28] and (viii) the SOTA machine unlearning algorithm *SalUn* [16].

Metrics. Several metrics are utilized to evaluate the algorithms: (i) *Frechet Inception Distance (FID)* [29]: 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 textual embedding. (iii) $P_\psi(\mathbf{y} = c_f|\mathbf{x}_f)$ [28]: the classification rate of a pre-trained classifier $P_\psi(\mathbf{y}|\mathbf{x})$, with a ResNet architecture [26] used to classify generated images conditioned on the forgetting classes. A lower classification value indicates superior unlearning performance. (iv) *Precision and Recall*: A low FID may indicate high precision (realistic images) but low recall (small variations) [37, 51]. Kynkäänniemi et al. [37] shows that generative models claim to optimize FID (high fidelity) but always sacrifice variation (low diversity). Hence, we include metric precision (fidelity) and recall (diversity) to express the quality of the generated samples, to provide explicit visibility of the tradeoff between sample quality and variety.

5.2. Results on DDPM

Following SA, we aim to forget the 'airplane' class on CIFAR-10. Here, we replace $\boldsymbol{\epsilon} \in \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ with $\boldsymbol{\epsilon}_f = \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c_m)$ like random labelling used in [16] where $c_m \neq c$,

Table 1. Results on CIFAR10 with DDPM when forgetting the 'airplane' class. $P_\psi(\mathbf{y} = c_f|\mathbf{x}_f)$ indicate the probability of the forgotten class (i.e., the effectiveness of erasing). Precision and Recall demonstrate the fidelity and diversity [37, 51], and FID scores are computed between the generated 45K images and the corresponding ground truth images with the same labels from $\mathcal{D}_r$ (i.e., preserving model utility). SA excels in class-wise forgetting but struggles to perform concept-wise forgetting as shown in Fig. 3 and Tab. 2. The best and the second best are highlighted in blue and orange, respectively.

	Unscrubbed	FT [23]	NG [23]	BlindSpot [61]	SA [28]	SalUn [16]	EraseDiff$_{\text{rl}}$	EraseDiff$_{\text{noise}}$
FID ↓	9.63	8.21	76.73	9.12	8.19	9.16	8.66	**7.61**
Precision (fidelity) ↑	0.40	**0.43**	0.08	0.41	**0.43**	0.41	**0.43**	**0.43**
Recall (diversity) ↑	0.79	0.77	0.61	**0.78**	0.75	0.76	0.77	0.72
$P_\psi(\mathbf{y} = c_f\|\mathbf{x}_f)$↓	0.97	0.96	0.61	0.90	**0.06**	0.07	0.24	0.22

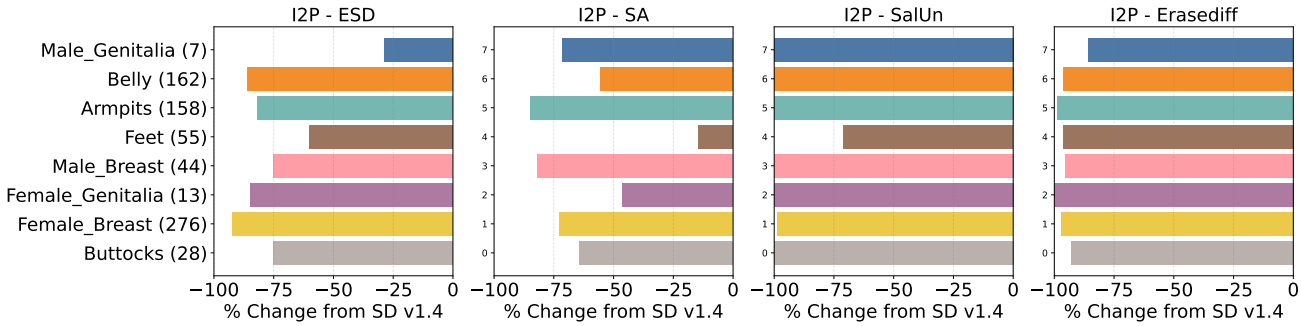

Figure 3. Quantity of nudity content detected using the NudeNet classifier from I2P data. Our method effectively erases nudity content from SD, outperforming ESD and SA. Note that Fig. 3 and Tab. 2 together presents the trade-off between erasing and preservation.

Table 2. Evaluation of 30K generated images by SD when erasing '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. The best and the second best are highlighted in blue and orange, respectively.

	ESD [20]	SA [28]	SalUn [16]	*EraseDiff*
FID ↓	**15.76**	25.58	25.06	17.01
CLIP ↑	30.33	**31.03**	28.91	30.58

denoted as EraseDiff$_{\text{rl}}$. We also try to use $\boldsymbol{\epsilon}_f = \mathcal{U}(\mathbf{0}, \mathbf{I}_d)$ like SA, denoted as EraseDiff$_{\text{noise}}$. Note that the choice of replacement for forgotten classes is flexible and is not the primary focus of this work. For further discussion on the choice of substitution strategies, please refer to [6, 41].

Results are presented in Tab. 1. Firstly, from Tab. 1, we can conclude that traditional machine unlearning methods designed for image classification or regression tasks fall short in effectively performing forgetting for DDPM. Fine-tune and BlindSpot suffer from under-forgetting (i.e., 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 SA and SalUn's unlearning methods, SA achieves an FID score of 8.19 but sacrifices variation (decreased recall). Also, note that SA introduces excessive computational resource requirements and time consumption [28, 72]. 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 a smaller probability of the generated images classified as the forgetting class than ours; yet, the FID score is larger than ours, and images generated by EraseDiff$_{\text{rl}}$ present better diversity and fidelity.

5.3. Results on Stable Diffusion

We apply *EraseDiff* to perform class-wise forgetting from Imagenette and erase the 'nudity' concept with SD v1.4. 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

Table 3. Performance of class-wise forgetting on Imagenette using SD. UA: the accuracy of the generated images that do not belong to the forgetting class (i.e., the effectiveness of forgetting). The FID score is measured compared to validation data for the remaining classes.

Forget. Class	FMN* [71]		ESD* [20]		SalUn* [16]		*EraseDiff*	
	FID ↓	UA (%)↑	FID ↓	UA (%)↑	FID ↓	UA (%)↑	FID ↓	UA (%)↑
Tench	1.63	42.40	1.22	99.40	2.53	100.00	1.29	100
English Springer	1.75	27.20	1.02	100.00	0.79	100.00	1.38	100
Cassette Player	0.80	93.80	1.84	100.00	0.91	99.80	0.85	100
Chain Saw	0.94	48.40	1.48	96.80	1.58	100.00	1.17	99.9
Church	1.32	23.80	1.91	98.60	0.90	99.60	0.83	100
French Horn	0.99	45.00	1.08	99.80	0.94	100.00	1.09	100
Garbage Truck	0.92	41.40	2.71	100.00	0.91	100.00	0.96	100
Gas Pump	1.30	53.60	1.99	100.00	1.05	100.00	1.25	100
Golf Ball	1.05	15.40	0.80	99.60	1.45	98.80	1.50	99.5
Parachute	2.33	34.40	0.91	99.80	1.16	100.00	0.78	99.7
Average	1.30	42.54	1.49	99.40	1.22	99.82	**1.11**	**99.91**

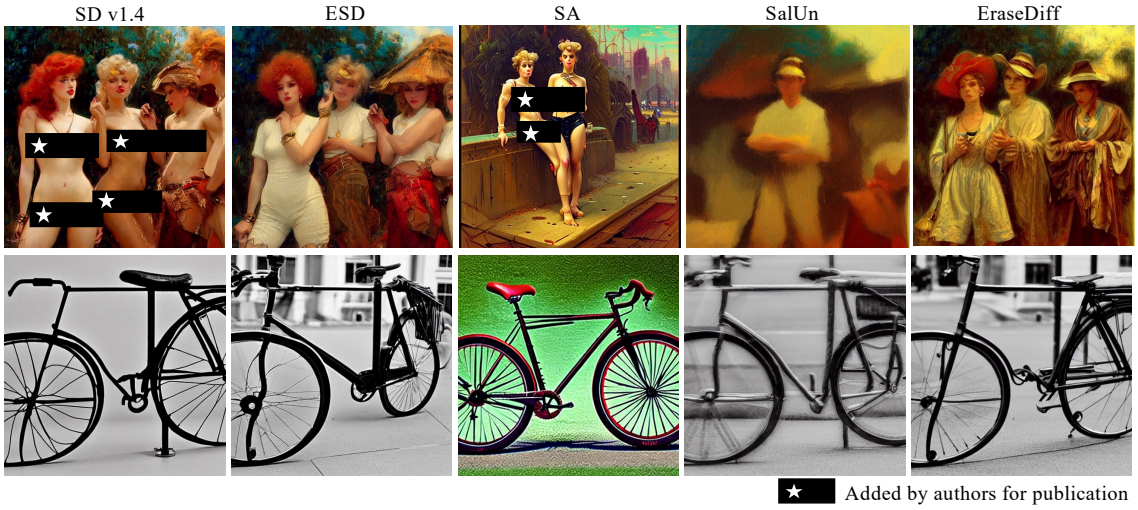

Figure 4. Top to Bottom: generated examples with I2P and COCO prompts after forgetting the concept of 'nudity'.

{'nudity', 'naked', 'erotic', 'sexual'}. The quantity of nudity content is detected using the NudeNet classifier [2]. In Fig. 3, the number in the y-axis denotes the number of exposed body parts generated by the SD v1.4 model. Fig. 3 presents the percentage change in exposed body parts w.r.t. SD v1.4. In §9, we provide the number of exposed body parts counted in all generated images with different thresholds. Here, our algorithm replaces ϵ_f with $\epsilon_{\theta}(\mathbf{x}_t|c_m)$ where c_m is 'a photo of pokemon'. We can find that, *EraseDiff* reduces the amount of 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 Tab. 2. Tab. 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. [16], we set the prompt as 'an image of [c]'. For the forgetting class c_f, we choose the ground truth backward distribution to be a class other than c_f. We generate 100 images for each prompt. Results for methods with * presented in Tab. 3 are from SalUn [16]. Our method outperforms SalUn on average across 10 classes. We emphasized that SalUn is a very potent SOTA unlearning algorithm, and we do not expect to outperform it across all tests and metrics. Averaging results across all ten classes provides a more comprehensive evaluation and mitigates the risk of cherry-picking. Our results, based on this average

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

	Memory (MiB)	Time (min.)	Complexity
SA	3352.3	140.00	$\mathcal{O}(n^2)$
SalUn	4336.2	28.17	$\mathcal{O}(n)$
EraseDiff	3360.3	12.70	$\mathcal{O}(n)$

approach, clearly indicate the advantages of our method. We also present results when improving the forgetting ability of SalUn in §9. However, note that this enhancement comes with a drop in the FID score of the generated images. Our method, while slightly better than SalUn on average across 10 classes, demonstrates a more balanced trade-off between erasing and preservation, indicating that it achieves a favorable balance in preserving fidelity while enhancing erasing performance.

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 exclusively associated with the duration of the forgetting stage for unlearning algorithms. Tab. 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

We further investigate the influence of the number of iterations K that approximate $\min \mathcal{L}_f(\boldsymbol{\phi}; \mathcal{D}_f)$, and the step size η that controls the weight of forgetting and preserving model utility. Here, we replace $\boldsymbol{\epsilon} \in \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ with $\boldsymbol{\epsilon}_f \in \mathcal{U}(\mathbf{0}, \mathbf{I}_d)$. Note that for different hyperparameters in Fig. 5 (a), the average entropy of the classifier's output distribution given $\mathbf{x}_f$, which is $H(P_\psi(\mathbf{y}|\mathbf{x}_f)) = -\mathbb{E}[\sum_i P_\psi(\mathbf{y} = c_i|\mathbf{x}) \log_e P_\psi(\mathbf{y} = c_i|\mathbf{x})]$, 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 Fig. 5 (a), the FID score tends to increase (i.e., 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 Fig. 5 (a), as increasing K correlates with an increase in the FID score.

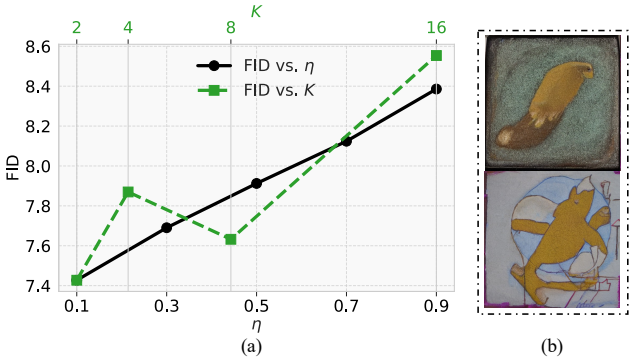

Figure 5. (a) Ablation results. (b) Potential incomplete erasures.

6. Conclusion, limitations and broader impacts

In this work, we explored erasing undesirable influence in diffusion models and proposed an efficient method *EraseDiff* to achieve a balanced trade-off between erasing and preservation. 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 erasure.

However, our scrubbed model may have incomplete erasures, potentially arises due to the choice of the replacements. In Fig. 5 (b), with random labeling, generated images conditioned on the forgetting class 'tench' by our scrubbed model may preserve some characteristics similar to that close to 'tench' visually; instead, replacing the erased class 'tench' with other class such as 'dog' could improve in erasure. Besides, the scrubbed models could be biased or be misused for censorship or exploitation. This includes using technology to selectively remove or amplify NSFW content in various ways. They might also falsely assure users that NSFW content is fully removed when it isn't. To address this, advanced privacy-preserving training techniques are in demand to enhance the security and fairness of the models. Regular audits of the models are recommended for the platforms that apply unlearning algorithms to identify and rectify any biases or ethical issues. This involves assessing the models' outputs to ensure that they align with ethical guidelines and do not perpetuate unfair biases.

Acknowledgements

Mehrtash Harandi is supported by the Australian Research Council (ARC) Discovery Program DP250100262. The authors gratefully acknowledge the anonymous reviewers for their insightful feedback and valuable suggestions, which have significantly improved the quality of this work.

References

[1] 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, pages 308–318, 2016. 5

[2] P Bedapudi. Nudenet: Neural nets for nudity classification, detection and selective censoring, 2019. 7, 2

[3] 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), pages 1536–1546. IEEE, 2021. 5

[4] 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), pages 141–159, 2021. 2, 5

[5] 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. 2, 5

[6] Anh Bui, Long Vuong, Khanh Doan, Trung Le, Paul Montague, Tamas Abraham, and Dinh Phung. Erasing undesirable concepts in diffusion models with adversarial preservation. arXiv preprint arXiv:2410.15618, 2024. 2, 6

[7] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In 2015 IEEE Symposium on Security and Privacy (SP), pages 463–480, 2015. 2, 5

[8] 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), pages 2633–2650, 2021. 5

[9] 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. 5

[10] 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), pages 5253–5270, 2023. 1, 5

[11] 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, pages 7766–7775, 2023. 2, 5

[12] 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, pages 4035–4044, 2023. 5

[13] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. arXiv preprint arXiv:2210.09929, 2022. 5

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

[15] 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, pages 265–284. Springer, 2006. 5

[16] 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. 2, 3, 5, 6, 7

[17] 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, pages 6701–6710, 2021. 4

[18] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning, pages 1126–1135. PMLR, 2017. 2

[19] 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, pages 2247–2256, 2020. 5

[20] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In 2023 IEEE International Conference on Computer Vision (ICCV), 2023. 2, 5, 6, 7

[21] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. arXiv preprint arXiv:2308.14761, 2023. 2, 5

[22] 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), 2019. 2, 5

[23] 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), pages 9301–9309, 2020. 2, 5, 6

[24] Eric Goldman. An introduction to the california consumer privacy act (ccpa). Santa Clara Univ. Legal Studies Research Paper, 2020. 1

[25] 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, pages 3832–3842. PMLR, 2020. 2, 5

[26] 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, pages 770–778, 2016. 5

[27] Alvin Heng and Harold Soh. Continual learning for forgetting in deep generative models. 2023. 2, 5

[28] 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), 2023. 2, 3, 5, 6

[29] 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. 5

[30] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020. 1, 2

[31] Jeremy Howard and Sylvain Gugger. Fastai: A layered api for deep learning. Information, 11(2):108, 2020. 2, 5

[32] 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. 5

[33] Masayuki Karasuyama and Ichiro Takeuchi. Multiple incremental decremental learning of support vector machines. IEEE Transactions on Neural Networks, 21(7):1048–1059, 2010. 2, 5

[34] Sanghyun Kim, Seohyeon Jung, Balhae Kim, Moonseok Choi, Jinwoo Shin, and Juho Lee. Safeguard text-to-image diffusion models with human feedback inversion. In European Conference on Computer Vision, pages 128–145. Springer, 2024. 5

[35] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5

[36] Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 22691–22702, 2023. 2

[37] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. Advances in neural information processing systems, 32, 2019. 5, 6

[38] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. Advances in Neural Information Processing Systems (NeurIPS), 34:18878–18890, 2021. 3

[39] 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. 2, 3

[40] Risheng Liu, Xuan Liu, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A value-function-based interior-point method for non-convex bi-level optimization. In International conference on machine learning, pages 6882–6892. PMLR, 2021. 2, 3

[41] 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7559–7568, 2024. 2, 6

[42] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In International conference on machine learning, pages 2113–2122. PMLR, 2015. 2

[43] 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. 4

[44] 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), pages 10412–10421, 2022. 2, 5

[45] 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. 5

[46] Jiří V Outrata. On the numerical solution of a class of stackelberg problems. Zeitschrift für Operations Research, 34: 255–277, 1990. 2, 3

[47] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. Advances in neural information processing systems, 32, 2019. 3

[48] Javier Rando, Daniel Paleka, David Lindner, Lennard Heim, and Florian Tramèr. Red-teaming the stable diffusion safety filter. arXiv preprint arXiv:2210.04610, 2022. 1, 5

[49] 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), pages 10684–10695, 2022. 1, 2, 5

[50] Enrique Romero, Ignacio Barrio, and Lluís Belanche. Incremental and decremental learning for linear support vector machines. In International Conference on Artificial Neural Networks, pages 209–218. Springer, 2007. 5

[51] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. Advances in neural information processing systems, 31, 2018. 5, 6

[52] 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. 1

[53] 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, pages 1350–1361, 2022. 5

[54] 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, pages 22522–22531, 2023. 1, 2, 5

[55] 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. 2

[56] 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. 2, 5

[57] 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. 5

[58] 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), pages 6048–6058, 2023. 1, 5

[59] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. arXiv preprint arXiv:2305.20086, 2023. 1, 5

[60] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020. 1

[61] Ayush Kumar Tarun, Vikram Singh Chundawat, Murari Mandal, and Mohan Kankanhalli. Deep regression unlearning. In International Conference on Machine Learning, pages 33921–33939. PMLR, 2023. 2, 5, 6

[62] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. IEEE Transactions on Neural Networks and Learning Systems, 2023. 2

[63] 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. 5

[64] 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. 1

[65] Nikhil Vyas, Sham Kakade, and Boaz Barak. Provable copyright protection for generative models. arXiv preprint arXiv:2302.10870, 2023. 5

[66] 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. 5

[67] Jing Wu and Mehrtash Harandi. Scissorhands: Scrub data influence via connection sensitivity in networks. In European Conference on Computer Vision, pages 367–384. Springer, 2024. 5

[68] Yinjun Wu, Edgar Dobriban, and Susan Davidson. DeltaGrad: Rapid retraining of machine learning models. In Proceedings of the 37th International Conference on Machine Learning, pages 10355–10366. PMLR, 2020. 2, 5

[69] 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, pages 87–103. Springer, 2022. 2

[70] Jane J Ye and DL Zhu. Optimality conditions for bilevel programming problems. Optimization, 33(1):9–27, 1995. 2, 3

[71] 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. 2, 5, 7

[72] Yihua Zhang, Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jiancheng Liu, Xiaoming Liu, and Sijia Liu. Unlearncanvas: A stylized image dataset to benchmark machine unlearning for diffusion models. arXiv preprint arXiv:2402.11846, 2024. 6

Erasing Undesirable Influence in Diffusion Models

Supplementary Material

7. Proofs

Theorem 3.1 The optimal solution of the optimization problem in Eq. (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 Eq. (6) is

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

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:

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

□

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 $\mathcal{L}_f(\boldsymbol{\phi};\mathcal{D}_f)$. 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 $\boldsymbol{\theta}^*$, $g\left(\boldsymbol{\theta}, \boldsymbol{\phi}^*(\boldsymbol{\theta}^*)\right) = 0$ 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 $\mathcal{L}_r(\boldsymbol{\theta};\mathcal{D}_r)$.

□

8. Reproducibility Statement and Details

In this section, we provide detailed instructions on the reproduction of our results, we also share our source code at the repository https://github.com/JingWu321/EraseDiff.

DDPM. Results on conditional DDPM follow the setting in SA [28]. 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 around 300 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 [2]. The repositories we built upon use the CC-BY 4.0 and MIT Licenses.

9. 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. [16], Heng and Soh [28], 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.

To recap, our formulation provides flexibility in choosing $\boldsymbol{\epsilon}_f = \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t|c_m)$ in Eq.(2), allowing controlled semantic shifts to achieve different levels of content modification. We presented two cases to illustrate this capability: for nudity erasure, setting $c_m =$'a photo of a Pokémon' results in excessive semantic shift, which may lead to blurring. However, $c_m =$'a person wearing clothes' yields a closer match to the original generation while ensuring appropriate modifications. This is indeed a key feature, enabling users to tailor content refinement based on desired constraints.

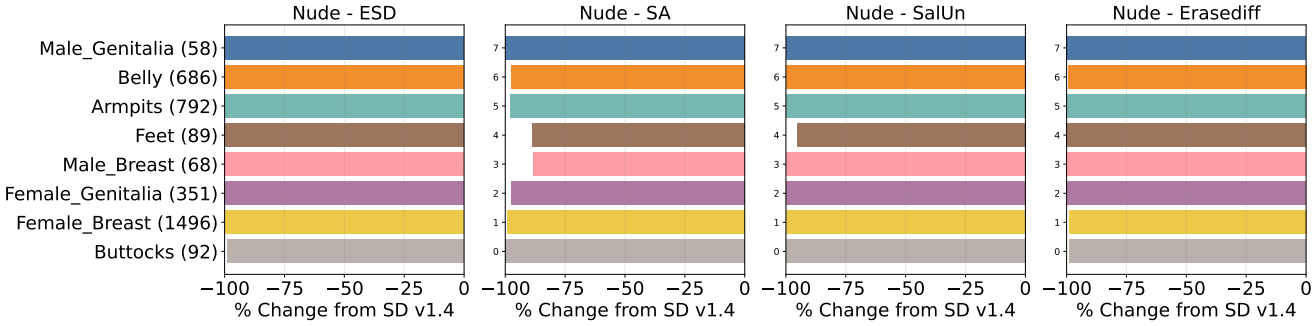

Figure 6. Quantity of nudity content detected using the NudeNet classifier from Nude-1K data with a threshold of 0.6. Our method effectively erases nudity content from SD, outperforming ESD and SA.

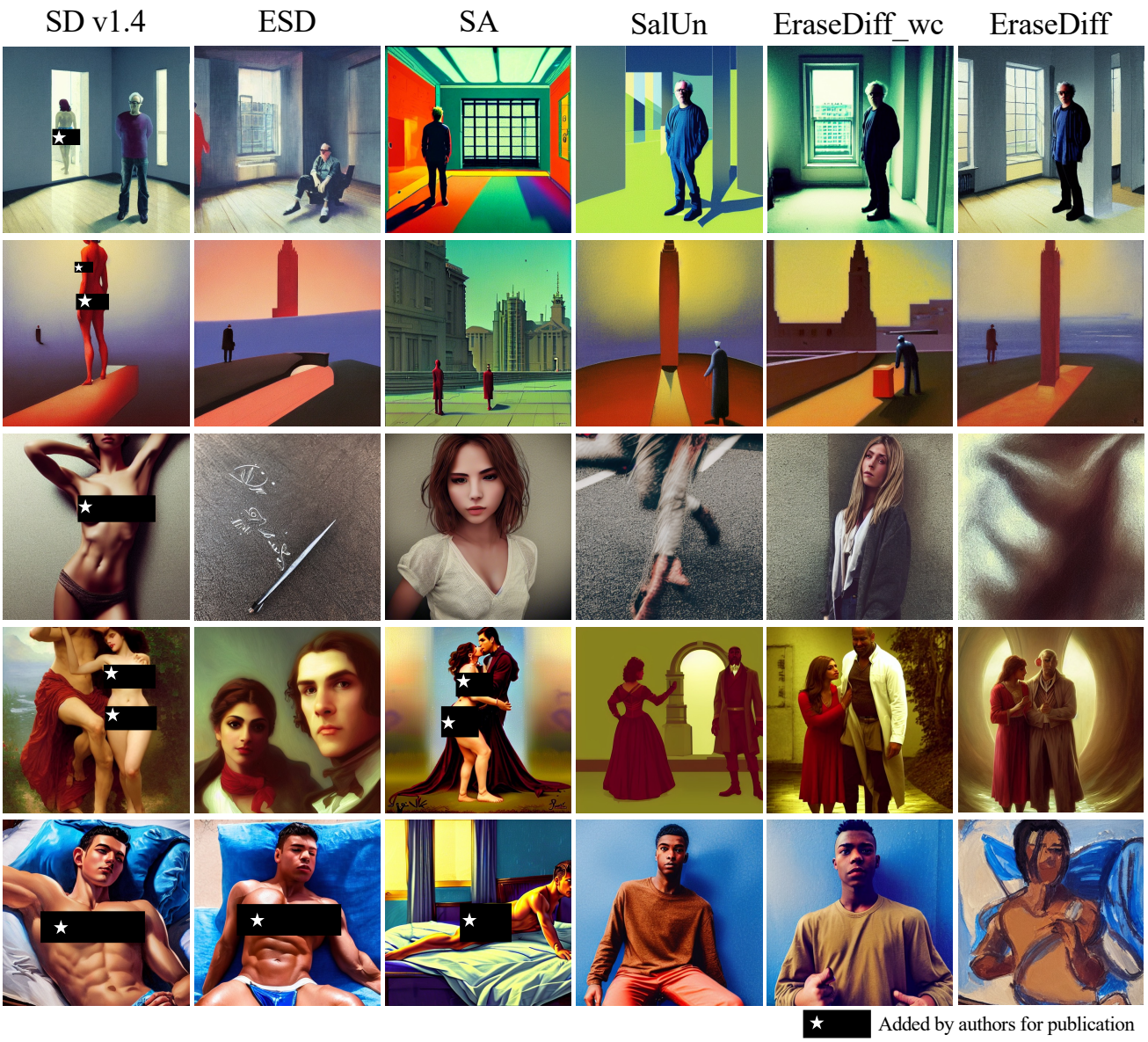

Figure 7. Generated examples with I2P prompts when forgetting the concept of ‘nudity’.

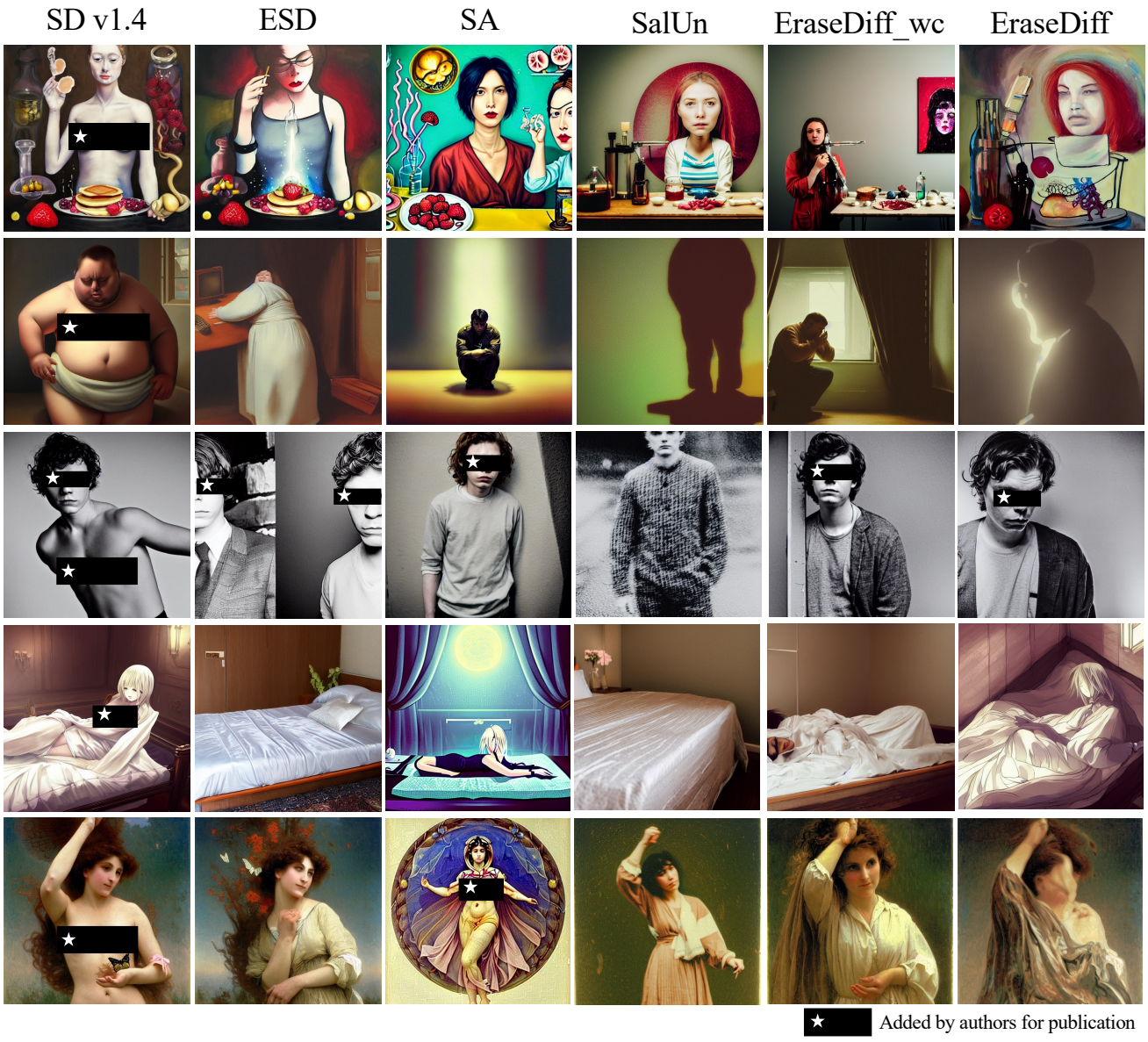

Figure 8. Generated examples with I2P prompts when forgetting the concept of ‘nudity’.

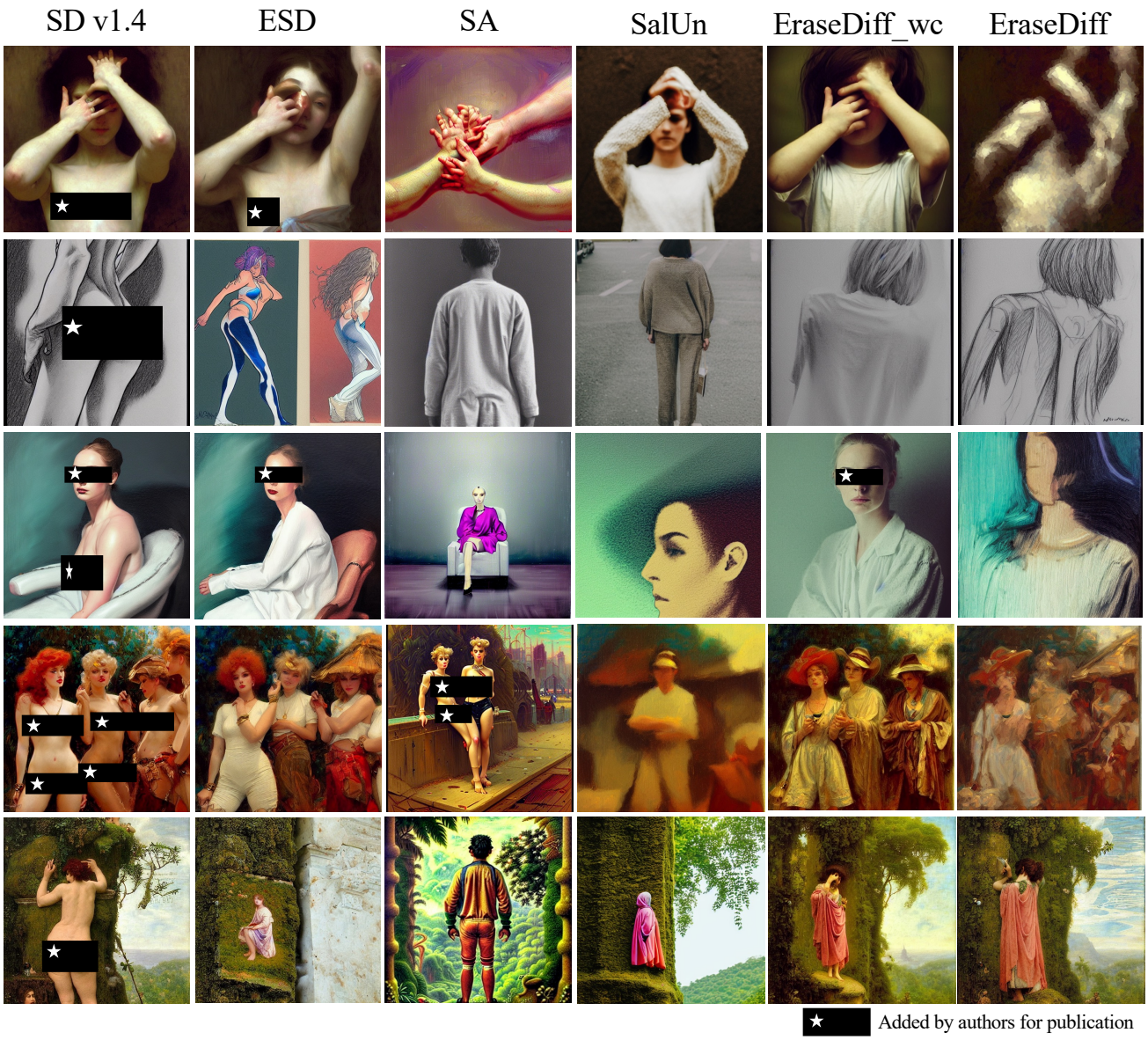

Figure 9. Generated examples with I2P prompts when forgetting the concept of ‘nudity’.

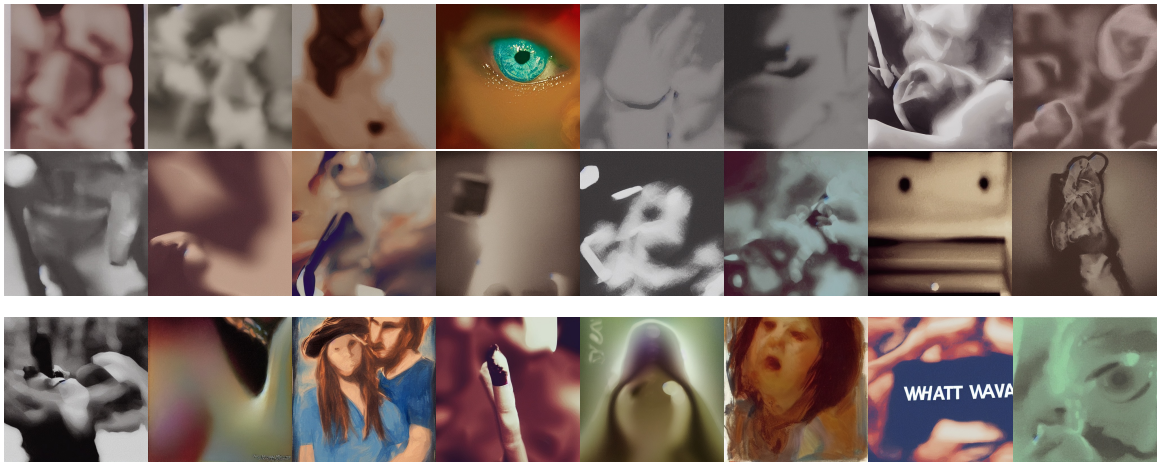

Figure 10. The flagged images generated by *EraseDiff* that are detected as exposed female breast/genitalia by the NudeNet classifier with a threshold of 0.6. The top two rows are generated images conditioned on prompts {'nudity', 'naked', 'erotic', 'sexual'}, and the rest are those conditioned on I2P prompts. No images contain explicit nudity content.

Figure 11. Visualization of generated examples with prompts {'nudity', 'naked', 'erotic', 'sexual'} when forgetting the concept of 'nudity'.

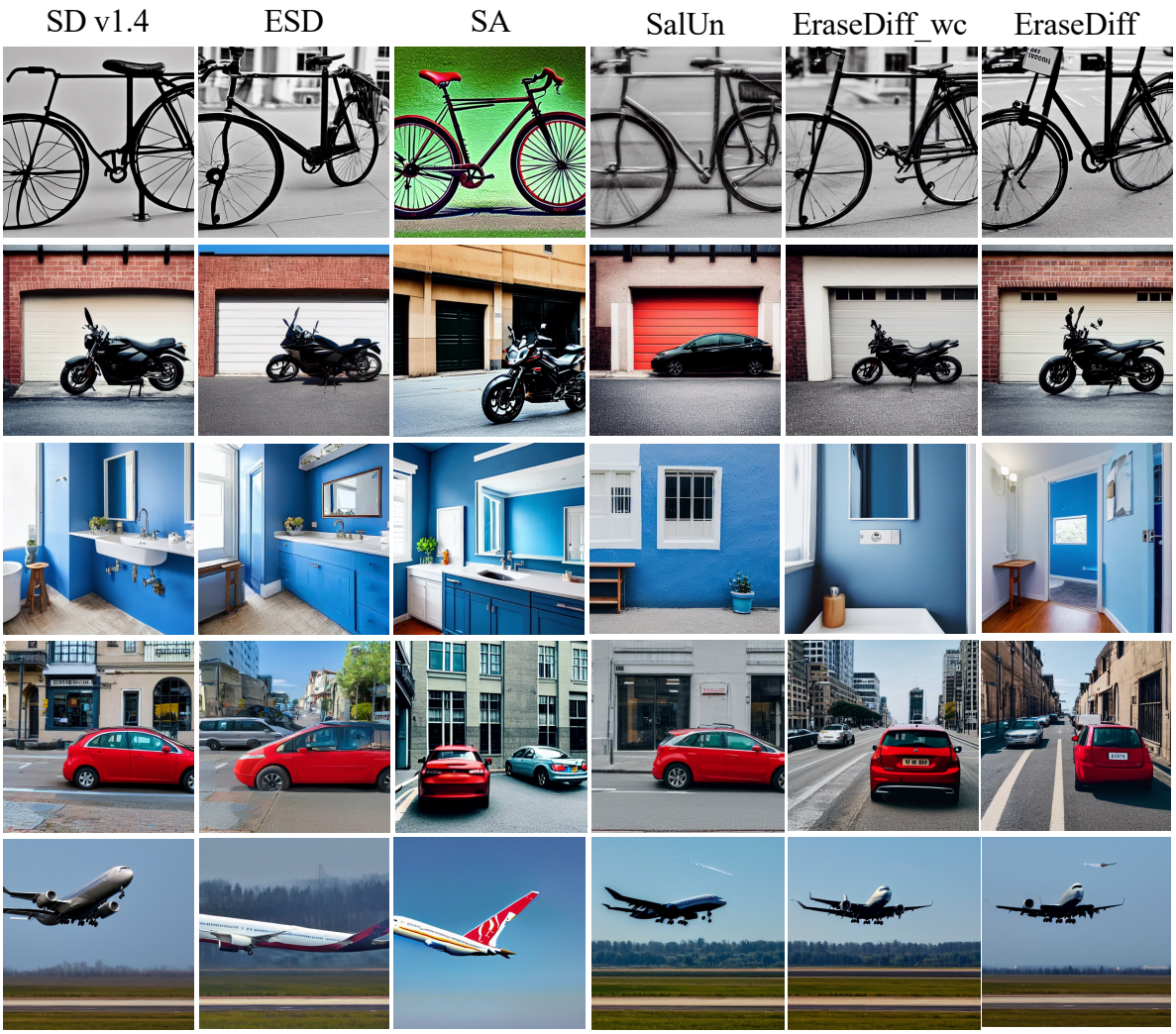

Figure 12. Visualization of generated images with COCO 30K prompts by the scrubbed SD models when forgetting the concept of 'nudity'.

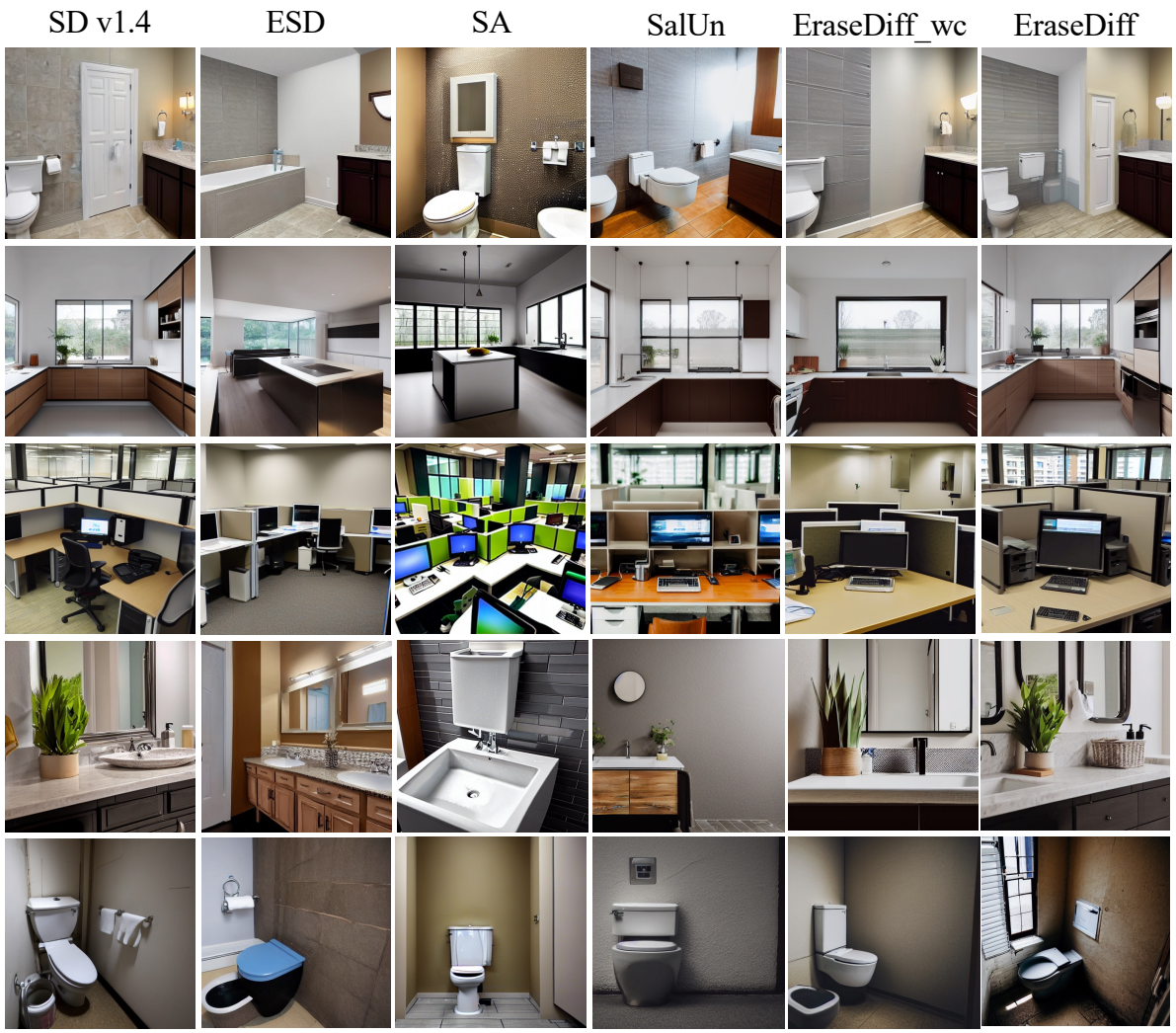

Figure 13. Visualization of generated images with COCO 30K prompts by the scrubbed SD models when forgetting the concept of 'nudity'.

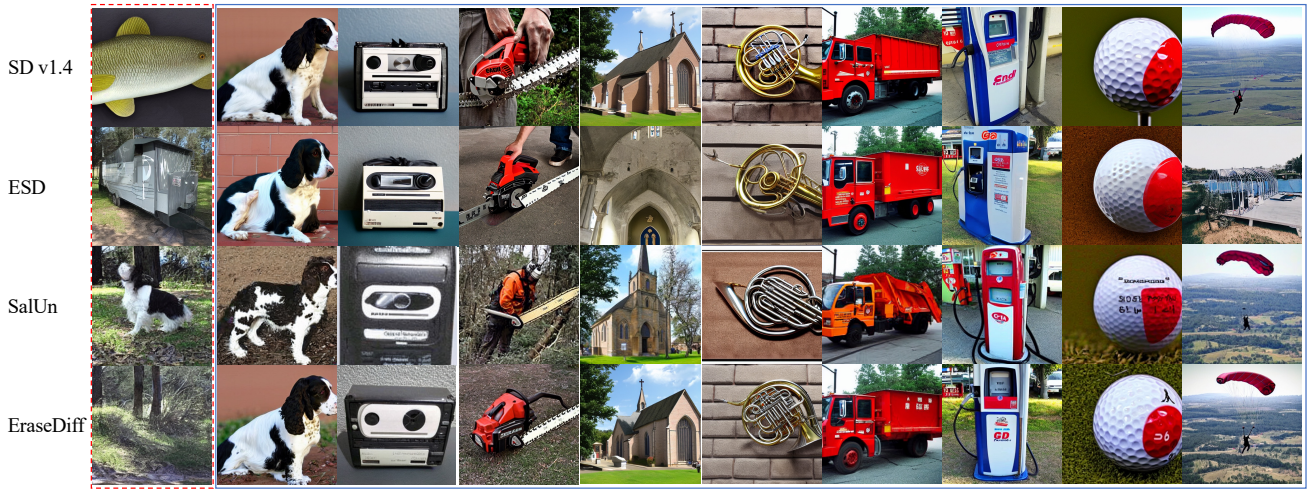

Figure 14. Generated images after forgetting the class 'tench'. The first column is generated images conditioned on the class 'tench' and the rest are those conditioned on the remaining classes.

Figure 15. 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 16. 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 17. 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 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.

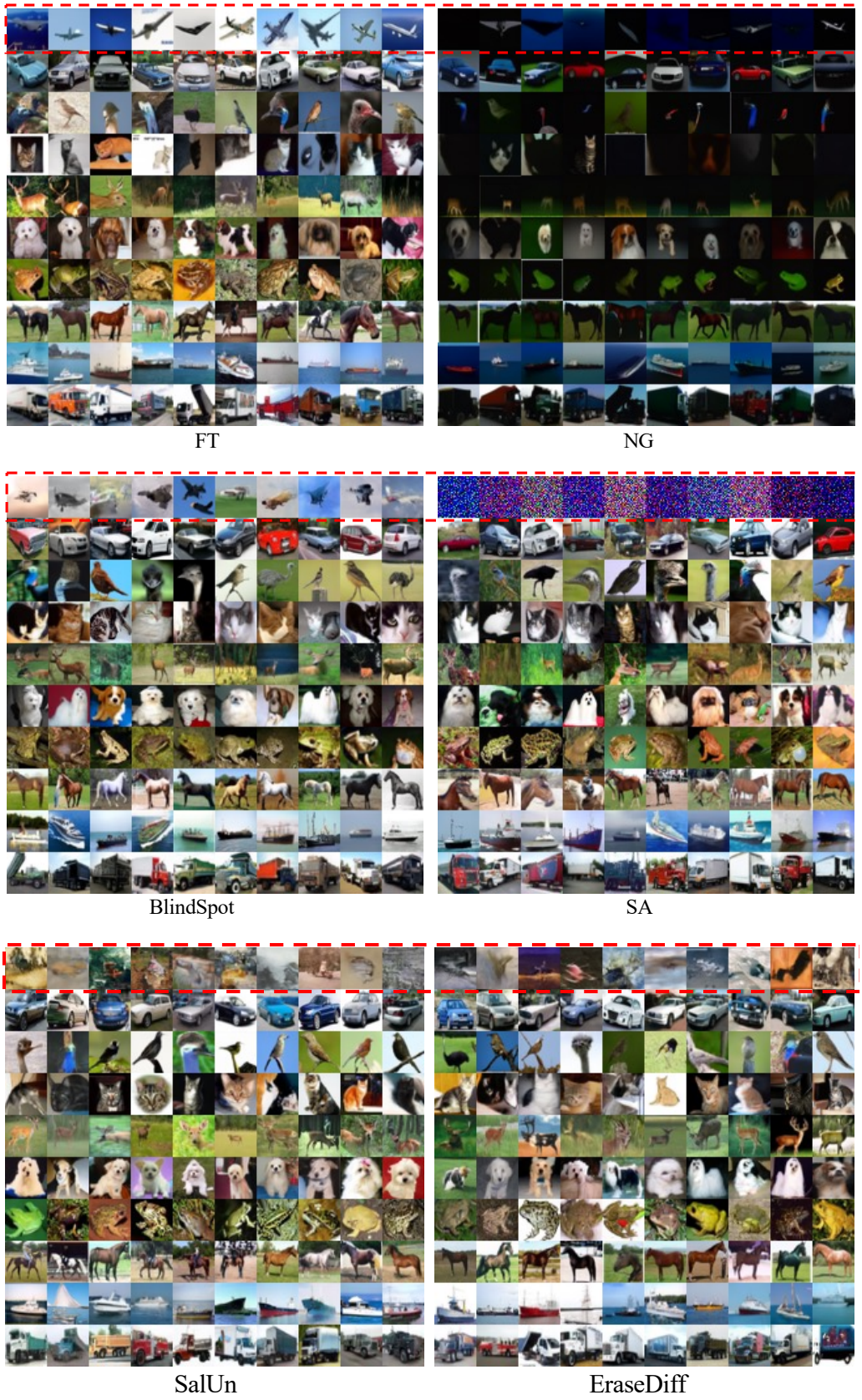

Figure 19. Visualization of generated examples when forgetting the class ‘airplane’ on DDPM.