UNLEARNING TRAINING DATA FROM DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models (DMs) have demonstrated remarkable generative capabilities in image generation but also pose privacy and copyright risks by memorizing and exposing training images. This concern is heightened by privacy regulations such as GDPR, which grant individuals the right to request the deletion of their data from AI models. Machine unlearning (MU) has been proposed to address this issue, as it enables the selective removal of specific training data from AI models. However, most existing MU methods for DMs primarily focus on unlearning at the class level—either by removing entire classes of data or class-specific features. In contrast, sample-level machine unlearning (SLMU), which targets the removal of individual training samples, remains an underexplored area. SISS is the pioneering work on SLMU for DMs. However, after careful investigation, we find that the evaluation metric used in SISS does not adequately assess unlearning performance. Moreover, under our proposed evaluation framework, SISS cannot achieve complete unlearning and presents significant degradation in generative performance. In this paper, we first define the objective of SLMU for DMs. Building on this definition, we introduce a quantitative evaluation framework for constructing benchmarks that compare different methods. Using this framework, we are the first to identify the *fake unlearning* phenomenon. Additionally, we propose a novel Sample-Level Machine Unlearning approach for Diffusion models, termed SMUD. SMUD alters the generative path of the targeted images, leading the DM to generate different images. Quantitative experimental results against baselines demonstrate that the proposed SMUD is the only method that can achieve SLMU without fake unlearning for both unconditional and conditional DMs.

1 INTRODUCTION

Diffusion models (DMs) have gained significant attention as powerful generative models. Trained on large-scale image datasets, DMs generate high-fidelity images that align closely with the training data distribution. DMs are broadly classified into two types: unconditional and conditional. Unconditional DMs generate high-quality images from Gaussian random inputs without additional information Ho et al. (2020); Nichol & Dhariwal (2021). In contrast, conditional DMs leverage auxiliary information to guide the generation process, enabling tasks such as text-to-image generation Rombach et al. (2022); Ramesh et al. (2022); Saharia et al. (2022b) and image-to-image translation Meng et al. (2022); Lugmayr et al. (2022); Saharia et al. (2022a).

However, alongside their impressive generative capabilities, DMs also raise privacy concerns; for instance, both unconditional and conditional DMs can generate duplicates of training images Wang et al. (2024b); Somepalli et al. (2023); Carlini et al. (2023); Chen et al. (2024b), leading to privacy breaches and copyright infringement. DMs present strong memorization, i.e., most generated data are duplicates of the training data when trained on small datasets Yoon et al. (2023); Baptista et al. (2025); Gu et al. (2023); Zhang et al. (2024b). Currently, some regulations, such as the General Data Protection Regulation (GDPR) GDP (2016) and the California Consumer Privacy Act (CCPA) CCP (2018), are addressing privacy and copyright risks. These regulations grant individuals the right to request the deletion of their data from a well-trained AI model.

Machine unlearning (MU) Bourtoule et al. (2021) has been proposed to remove the training data from a trained AI model to ensure privacy and copyright compliance. MU solutions can be catego-

rized into exact unlearning and approximate unlearning Thudi et al. (2022). Exact unlearning Bourtoule et al. (2021); Yan et al. (2022) removes the training data from the model through algorithmic-level retraining and applies ensemble models for acceleration. Approximate unlearning Mehta et al. (2022); Golatkar et al. (2020b); Neel et al. (2021) aims to minimize the influence of targeted data points to an acceptable level rather than completely removing them. However, the above-mentioned unlearning methods are designed for classification models and unsuitable for DMs. Details about MU for classification models are available in Appendix A.1. On the other hand, most existing MU methods for DMs are primarily focus on class-level MU, such as unlearning an entire class of data or a class of features Zhang et al. (2024c;a); Li et al. (2024a); Fan et al. (2024); Fuchi & Takagi (2024); Gandikota et al. (2023). These methods cannot solve the finer-grained sample-level machine unlearning (SLMU) since they require a conditioning input and unlearn all features related to the input. Details about MU for DMs are available in Appendix A.2.

Another challenge of SLMU for DMs lies in defining appropriate evaluation metrics that measure how well the DMs unlearn the targeted samples. For class-level MU in DMs, we can evaluate the performance of MU methods by verifying whether the unlearned DMs generate the targeted classes or features, but this evaluation is not suitable for SLMU. On the other hand, evaluation metrics for SLMU in classification models are usually based on the model's output Chundawat et al. (2023); Fan et al. (2024); Chen et al. (2023a); Foster et al. (2024); Kurmanji et al. (2024); Liu et al. (2024). However, these metrics are unsuitable for DMs, as the output of a DM is a random three-dimensional vector rather than a deterministic classification vector. Furthermore, the distributions of the generated images before and after unlearning may appear indistinguishable Stadler et al. (2022); Yuan et al. (2024a;b) since the targeted unlearning data are in-distribution for SLMU. SISS Alberti et al. (2025) is the pioneering work in SLMU for DMs. However, after careful investigation (detail in Section 3.2), we find that the evaluation metric used in SISS is inadequate. Besides, SISS is built on a problematic assumption, i.e., fine-tuning on $X \setminus A$ can unlearn the unlearning set A from the pretrained model, as Baseline-F in Fig. 7 and 14 in Appendix cannot achieve complete unlearning. Under our proposed evaluation framework, SISS fails to achieve complete unlearning and exhibits significant degradation in generative performance. To address these research gaps, this paper introduces a novel method—Sample-level Machine Unlearning for Diffusion models, termed as SMUD. SMUD alters the generative path of the targeted unlearning images, causing the DM to generate different images from those initially presented. Besides, this paper proposes a quantitative evaluation framework based on the memorization property of DMs, which can be used to construct a benchmark and thus provides a foundation for future research.

In summary, the paper makes the following contributions:

- We propose a ***novel quantitative evaluation framework*** for SLMU in DMs, leveraging DMs' memorization property, which can be used to construct a benchmark and thus provides a foundation for future research.
- To the best of our knowledge, we are the *first* to observe the *fake unlearning* phenomenon in machine unlearning and incorporate it into the proposed evaluation framework.
- We propose ***a novel SLMU method***, SMUD, which intentionally changes the generation path of the targeted images to avoid generating them.
- We provide a comprehensive evaluation of the proposed SMUD. Quantitative results against four baselines demonstrate that our proposed SMUD is the only method to achieve SLMU without fake unlearning for both unconditional and conditional DMs.

2 Preliminaries for Diffusion Models

The DMs introduced in this section are based on DDPM Ho et al. (2020). DDPM operates in two stages, i.e., the forward and reverse processes. The forward process starts from clean data $\mathbf{x}_0$ and iteratively adds Gaussian noise to the data for T steps until the data $\mathbf{x}_T$ becomes nearly indistinguishable from pure Gaussian noise. Given a forward process step t, $\mathbf{x}_t$ can be calculated by,

$$\mathbf{x}_t\left(\mathbf{x}_0, \boldsymbol{\epsilon}\right) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \text{ for } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

where $\bar{\alpha}_t$ is a pre-defined parameter and $t \in \{1, 2, \cdots, T\}$. The reverse process starts from pure Gaussian noise $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively denoises the data with the estimated noise, $\hat{\boldsymbol{\epsilon}} =$

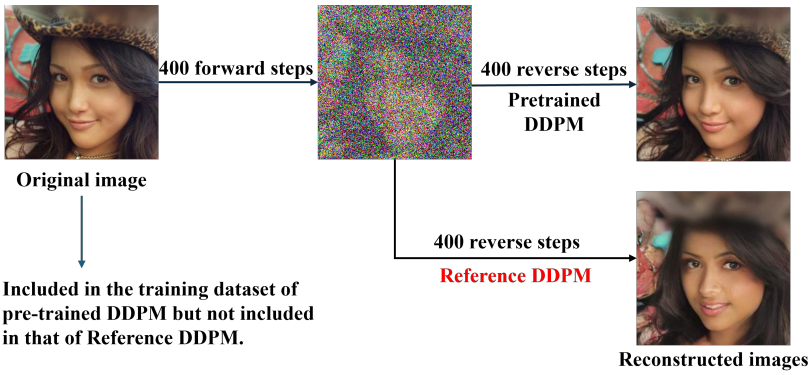


Figure 1: Demonstration of memorization of DMs where the original image is not included in the training dataset of reference DDPM but included in that of pre-trained DDPM.

$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t, t)$, where $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is a trainable approximator, until getting $\hat{\mathbf{x}}_0$. Given $\hat{\mathbf{x}}_t$, $\hat{\mathbf{x}}_{t-1}$ can be calculated by,

$$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\hat{\mathbf{x}}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_t, t\right)\right) + \sigma_t \mathbf{z}_t, \tag{2}$$

where $t \in \{1, 2, \cdots, T\}$, $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and σ_t and α_t are pre-defined parameters. To make the distribution of $\hat{\mathbf{x}}_0$ similar to that of $\mathbf{x}_0$, the deep learning-based approximator $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is optimized according to the following loss function,

$$L(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}) = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right]. \tag{3}$$

On the other hand, conditional DDPM incorporates conditioning input into the reverse process enabling more targeted and controlled generation. The forward process of conditional DDPM is the same as **Eq.**(1). But it considers the conditioning input $\mathbf{c}$ to estimate the noise at the step t of the reverse process. Given $\hat{\mathbf{x}}_t$ in the reverse process of conditional DDPM, $\hat{\mathbf{x}}_{t-1}$ can be calculated by,

$$\begin{aligned} \hat{\boldsymbol{\epsilon}} &= \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_t, t, \emptyset\right) + \beta\left(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_t, t, \mathbf{c}\right) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_t, t, \emptyset\right)\right), \\ \hat{\mathbf{x}}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\left(\hat{\mathbf{x}}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\epsilon}}\right) + \sigma_t \mathbf{z}_t, \end{aligned} \tag{4}$$

where $\mathbf{c}$ is the conditioning input, $\emptyset$ is the feature for the null condition that is usually a zero vector, and β is a scalar for the conditional scale. Accordingly, the loss function of conditional DDPM is calculated by,

$$L_c(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}) = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \boldsymbol{c}\right)\right\|^2\right]. \tag{5}$$

3 Evaluation Framework

In this section, we first define the objectives of SLMU in DMs and then present the proposed evaluation framework based on this definition.

3.1 Objective Definition

The objective of SLMU for general machine learning models has been defined in Bourtoule et al. (2021) based on the distribution of model parameters. However, the distribution of model parameters is hard to measure for evaluation purposes. Moreover, we need to train multiple DMs to estimate the parameter distribution, which is unrealistic considering the computational requirements of training a DM. On the other hand, the provider of image generation service usually keeps the model parameters private and releases an API for users to generate synthetic images using a well-trained DM. In this scenario, potential privacy breaches and the impact of individual data are primarily manifested through the generated images. In this regard, Definition 1 defines the objective of SLMU for DMs based on the distribution of generated images.

Definition 1 *Let $\boldsymbol{\theta}_p$ denote a pretrained DM that is trained on dataset D. Select a subset D_u from D as the unlearning set and retain set $D_r = D \setminus D_u$. Given an SLMU mechanism $\mathcal{M}$, the unlearned DM $\boldsymbol{\theta}_u$ is obtained by $\boldsymbol{\theta}_u = \mathcal{M}(\boldsymbol{\theta}_p, D_r, D_u)$. Let P_u denote the distribution of the images generated by $\boldsymbol{\theta}_u$. The objectives of $\mathcal{M}$ are to (i) increase the similarity between P_u and the distribution of the retain set and (ii) erase the memorization of D_u in $\boldsymbol{\theta}_p$.*

Note: Same as Alberti et al. (2025), the size of the unlearning set D_u is assumed to be much smaller than that of the training dataset D. As a result, D_r has a similar distribution to D.

3.2 Proposed Evaluation Framework

In this Section, we first present challenges in evaluation metrics for SLMU and describe the problems with the evaluation metrics proposed in Alberti et al. (2025). Then, the proposed quantitative evaluation framework is introduced.

Challenges in evaluation metrics. As outlined in Definition 1, SLMU in DMs has two primary objectives. Objective (i) can be effectively evaluated using the FID Heusel et al. (2017). However, objective (ii) presents a significant challenge—there is no clear method to definitively determine if the memorization of D_u has been fully unlearned. The main reason is that the distribution of the training dataset D is indistinguishable from that of the retain set D_r under practical metrics, as the size of the unlearning set D_u is small and its samples are in-distribution. Therefore, if objective (i) is met, the distributions of generated data before and after unlearning are indistinguishable. This phenomenon has also been observed in Stadler et al. (2022); Yuan et al. (2024a;b).

Problems with existing evaluation metrics. In Fig. 1, the original image $\mathbf{x}_{ori}$ is part of the training dataset D for the pretrained DDPM, while the reference DDPM is trained on $D \setminus \{\mathbf{x}_{ori}\}$. In this setup, the reference DDPM serves as the ideal unlearned model for the pretrained DDPM, where the unlearning set $D_u = \{\mathbf{x}_{ori}\}$. To evaluate the memorization, we generate a noised image with 400 forward steps, which retains partial information from the original image. **Note** that after 1000 forward steps, the image becomes pure Gaussian noise and is non-reconstructable. We then conduct 400 reverse steps to reconstruct the image using both the pretrained and reference DDPMs. As depicted in Fig. 1, the pretrained DDPM reconstructs an image that is nearly identical to the original, while the reference DDPM reconstructs an image that slightly differs from the original. The authors of SISS Alberti et al. (2025) argue that a larger difference between the original and reconstructed images indicates more effective unlearning. However, this definition is problematic. As shown in Fig. 1 and Fig. 3, although the reference DDPM has never seen the original image during training, it can reconstruct an image that closely resembles the original image. Since the reference model is the ideal unlearned model, we cannot conclude that a larger difference between the original and reconstructed images necessarily means more effective unlearning.

Proposed quantitative evaluation metric. We use the strong memorization of DMs for evaluation instead of the process demonstrated in Fig. 1 used in Alberti et al. (2025). First, a DM is pretrained on a small dataset D. Second, generate a synthetic dataset $\hat{D}_p$ with the pretrained DM. Third, select the replicates of D's data from the synthetic dataset $\hat{D}_p$, using the same method as Yoon et al. (2023); Gu et al. (2023). The unlearning set D_u is constructed by the N most memorized training data. Then, we unlearn the pretrained DM and generate a synthetic dataset $\hat{D}_u$ with the unlearned DM. Last, we select (using the same method as Yoon et al. (2023); Gu et al. (2023)) and count the duplicates of D_u's data from $\hat{D}_u$ as the quantitative evaluation metric, which is termed as Number of Duplicates of the Unlearning Set (NDUS). Smaller NDUS means better unlearning.

Fake unlearning. In our preliminary experiments, we observed that initially unlearned DMs, which do not generate duplicates of images from the unlearning set, start to generate such duplicates after fine-tuning on the retain set. This phenomenon is termed as fake unlearning. The unlearned DM initially avoids generating duplicates due to the performance degradation caused by the unlearning process, and fine-tuning the unlearned model on the retain set can recover the generative performance. Fake unlearning indicates that the DM does not completely forget the unlearning data. We incorporate this fake unlearning in our proposed evaluation framework by measuring NDUS after fine-tuning the unlearned DM on the retain set. The overall evaluation framework is summarized in Appendix B.

Rationale behind NDUS. According to Theorem 4.3 in Baptista et al. (2025), during the reverse process of DMs, if any intermediate result enters the Voronoi cell of a training sample, the generation trajectory will converge to that sample. Experimental results in Baptista et al. (2025) also show that DMs always generate duplicates of training data when the model has enough parameters. This finding is consistent with other works Yoon et al. (2023); Gu et al. (2023); Zhang et al. (2024b). Based on this theorem, if a DM stops generating previously memorized data, we can infer that it has unlearned that training sample. However, as discussed earlier, fake unlearning can occur due to a decline in generative performance during the unlearning process. To address this, we fine-tune the unlearned DM on the retain set to recover its generative capabilities and then check whether it generates the previously memorized data.

4 Sample-Level Machine Unlearning for Diffusion Models

In this section, we first provide a detailed introduction to SMUD for unconditional DMs and then briefly describe SMUD for conditional DMs, which is largely identical to the unconditional case.

4.1 SMUD for Unconditional DMs

To achieve the objectives of SLMU, we first define the **noised reverse process** for unconditional DMs as follows,

$$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\hat{\mathbf{x}}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \left(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t, t) + \gamma \boldsymbol{\epsilon}' \right) \right) + \sigma_t \mathbf{z}_t, \tag{6}$$

where $\gamma \in (0, \infty)$ is a coefficient controlling the noise amplitude, $\boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and other parameters and variables are the same as **Eq.**(2). **Eq.**(6) can be rewritten as,

$$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\hat{\mathbf{x}}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \left(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t, t) \right) \right) + \frac{\gamma(1-\alpha_t)}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}} \boldsymbol{\epsilon}' + \sigma_t \mathbf{z}_t. \tag{7}$$

The summary of the last two terms in **Eq.**(7), i.e., $\frac{\gamma(1-\alpha_t)}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}} \boldsymbol{\epsilon}' + \sigma_t \mathbf{z}_t$, follows a Gaussian distribution with mean value of $\mathbf{0}$ since both $\boldsymbol{\epsilon}'$ and $\mathbf{z}_t$ follows $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Therefore, the noised reverse process **Eq.**(7) can be seen a standard reverse process **Eq.**(2) with a larger σ_t. According to the analysis in Kynkäänniemi et al. (2019), if γ is properly chosen (neither too large nor too small), and given the same $\hat{\mathbf{x}}_T$, well-trained $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$, and $\mathbf{z}_t$, the noised and standard reverse processes will produce different images. This has been experimentally validated in Section 5.1.

To achieve SLMU, we use the noised reversed process to fine-tune the pretrained DM on the unlearning set. Specifically, we add a Gaussian noise to $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ and use the result as the label to optimize $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ when the input $\mathbf{x}_0$ is sampled from unlearning set D_u. The unlearning loss L_u for unconditional DMs is calculated as,

$$L_u(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}) = \mathbb{E}_{t, \mathbf{x}_0 \in D_u, \boldsymbol{\epsilon} \text{ and } \boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\| \boldsymbol{\epsilon}'_{\boldsymbol{\theta}}(\mathbf{x}_t, t) + \gamma \boldsymbol{\epsilon}' - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \|^2], \tag{8}$$

where $\boldsymbol{\epsilon}'_{\boldsymbol{\theta}}$ is a copy of $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ and not optimized during unlearning, $\gamma \in (0, \infty)$ controls the noise amplitude, $\boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and other parameters are the same as **Eq.**(3). Optimizing $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ by minimizing **Eq.**(8) alters the generation path of the unlearning set images to other images and thus achieve SLMU. Moreover, this unlearning loss does not significantly affect the generation performance, as the altered images remain within the distribution of the training dataset.

On the other hand, the distribution of the images generated by the unlearned model is required to be similar to the distribution of the retain set D_r as discussed in Definition 1. To achieve this objective, when the input $\mathbf{x}_0$ is sampled from the retain set, we apply the original loss functions of unconditional DMs as the retain loss L_r, i.e.,

$$L_r(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}) = \mathbb{E}_{t, \mathbf{x}_0 \in D_r, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t) \|^2]. \tag{9}$$

We optimize the approximator by minimizing the unlearning loss **Eq.**(8) every N_{interval} optimization steps to facilitate unlearning while by minimizing the retain loss **Eq.**(9) at each optimization step to preserve the model's generative capability.

4.2 SMUD FOR CONDITIONAL DMS

Similar to the unconditional DDPM, the noised reverse process for conditional DMs is defined as,

$$\begin{aligned}\hat{\boldsymbol{\epsilon}} &= \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_t, t, \emptyset\right) + \beta\left(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_t, t, \mathbf{c}\right) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\hat{\mathbf{x}}_t, t, \emptyset\right)\right), \\ \hat{\mathbf{x}}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\left(\hat{\mathbf{x}}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}(\hat{\boldsymbol{\epsilon}} + \gamma\boldsymbol{\epsilon}')\right) + \sigma_t \mathbf{z}_t,\end{aligned} \tag{10}$$

where $\gamma \in (0, \infty)$ is a coefficient controlling the noise amplitude and $\boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Similar to the unconditional case, the noised reverse process **Eq.**(10) can be seen a standard reverse process **Eq.**(4) with a larger σ_t. According to the analysis in Kynkäänniemi et al. (2019), if γ is properly chosen (neither too large nor too small), and given the same $\hat{\mathbf{x}}_T$, well-trained $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$, and $\mathbf{z}_t$, the noised and standard reverse processes will produce different images that follow a distribution similar to the training dataset, which has been validated experimentally in Section 5.1.

Unlike unconditional DMs, unlearning set images' information is encoded in $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ when the conditions are either $\mathbf{c}$ (the ground-truth condition) or $\emptyset$ (the null condition). To simulate the reverse process **Eq.**(10), the unlearning loss, L_u, for conditional DMs is calculated by,

$$\begin{aligned}&\mathbf{y}_{\mathbf{c}} = (\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{c}), \quad \mathbf{y}_{\emptyset} = (\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \emptyset), \\ &L_u(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}) = \mathbb{E}_{t, \mathbf{x}_0 \in D_u, \boldsymbol{\epsilon} \text{ and } \boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}[\|\boldsymbol{\epsilon}'_{\boldsymbol{\theta}}(\mathbf{y}_{\mathbf{c}}) + \gamma\boldsymbol{\epsilon}' - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{y}_{\mathbf{c}})\|^2 + \|\boldsymbol{\epsilon}'_{\boldsymbol{\theta}}(\mathbf{y}_{\emptyset}) + \gamma\boldsymbol{\epsilon}' - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{y}_{\emptyset})\|^2].\end{aligned} \tag{11}$$

Similar to the unconditional case, we need to optimize $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ according to the retain loss to maintain the generative capability of the DM. The retain loss, L_r, for conditional DMs is calculated as,

$$L_r(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}) = \mathbb{E}_{t, \mathbf{x}_0 \in D_r, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{c})\|^2]. \tag{12}$$

The pipeline of SMUD is summarized in Appendix C.

5 EXPERIMENTS

In this section, we quantitatively evaluate SMUD with the proposed evaluation framework. Then, we qualitatively evaluate SMUD trained on a large dataset as demonstrated in Fig. 1.

5.1 EVALUATION OF NOISED REVERSE PROCESSES

To evaluate the effectiveness of the noised reverse process **Eq.**(6) for unconditional DMs, we apply the pretrained $\boldsymbol{\theta}_p$ to generate images using standard reverse process, i.e., $\gamma = 0$ in **Eq.**(6), and noised reverse process, i.e., $\gamma > 0$ in **Eq.**(6). As shown in the left column of Fig. 2, when γ is small, e.g., $\gamma = 0.05$, the images generated by the standard and noised reverse processes are almost the same. When γ becomes larger, e.g., $\gamma = 0.1$, the images generated by the standard and noised reverse processes become more different. Note that $\mathbf{z}_t$ and $\hat{\mathbf{X}}_T$ in **Eq.**(2) keep the same by using the same random seed across different γ values. These results demonstrate that injecting Gaussian noise into $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t, t)$ can alter the generated samples while ensuring that they remain consistent with the training distribution. Consequently, the proposed SMUD preserves the generative capability of the model after unlearning. Similar observations are obtained for conditional DMs, as illustrated in the right column of Fig. 2.

5.2 BASELINES AND EVALUATION METRICS.

In this paper, we consider four baselines. The first baseline is training the DM on D_r from scratch, referred to as Baseline-R. In the field of continual learning Wang et al. (2024a), it has been observed that fine-tuning a pre-trained deep learning model on new data can degrade its performance on previously learned data. Building on this property, the second baseline involves fine-tuning the pre-trained DM on the retain dataset D_r, termed as Baseline-F. The third baseline is based on gradient ascent Huang et al. (2024), termed as Baseline-GA. Although the method in Huang et al. (2024) is initially proposed for class-level MU for DMs, it can be adapted to SLMU. The fourth baseline is SISS Alberti et al. (2025), which is the pioneering work to address SLMU in DMs.

For evaluation, we first use the unlearned DM to generate a synthetic dataset $\hat{D}_u$ of the same size as the pre-training dataset. We then compute the FID (denoted as FID_M) between the synthetic dataset

(a) $\gamma = 0$ (b) $\gamma = 0$

(c) $\gamma = 0.05$ (d) $\gamma = 0.05$

(e) $\gamma = 0.1$ (f) $\gamma = 0.1$

Figure 2: Images generated by the pretrained $\boldsymbol{\theta}_p$ with different γ values in the noised reverse process **Eq.**(6) (left column) and **Eq.**(10) (right column).

$\hat{D}_u$ and retain set D_r to evaluate objective (i) of Definition 1. Comparing the synthetic dataset $\hat{D}_u$ and unlearning set D_u, we select duplicates of D_u's data from $\hat{D}_u$ using the method in Yoon et al. (2023); Gu et al. (2023) and count the duplicates as NDUS (denoted as NDUS_M) to evaluate the objective (ii) of Definition 1. Finally, to detect fake unlearning, we fine-tune the unlearned DM on D_r and compute FID and NDUS again (denoted as FID_F and NDUS_F, respectively).

5.3 Experiments for Unconditional DMs

We apply the unconditional DDPM proposed in Nichol & Dhariwal (2021) with a linear noise schedule and 1K diffusion steps. We randomly select 2K images from each of the five classes in the CIFAR-10 dataset—automobile, airplane, bird, cat, and deer—as five separate training sets. We pre-train the DDPM for 1M steps with the mini-batch size 128 on a training set and obtain the pre-trained DDPM $\boldsymbol{\theta}_p$, and then use it to generate a synthetic dataset $\hat{D}_p$ with 2K images. Then we select 16 images in D, which have the most duplicates in $\hat{D}_p$, to construct the unlearning set D_u and the remaining 1984 images of D construct the retain set D_r. We set $N_{\text{interval}} = 5$ and $\gamma = 1.0$.

Table 1 presents a performance overview of the proposed SMUD and two baselines on unlearning five CIFAR-10 classes separately. For SMUD and Baseline-GA, the DMs are unlearned for 4K steps and subsequently finetuned for 4K steps to assess fake unlearning. For SISS, fewer unlearning steps are applied, as excessive steps severely degrade generation quality (see Appendix D); the unlearned model is then fine-tuned for 4K steps. As shown in Table 1, SMUD achieves the best generative performance after unlearning without exhibiting fake unlearning. In contrast, both Baseline-GA and SISS significantly reduce the generative quality and display fake unlearning. Finetuning the DMs unlearned by Baseline-GA and SISS results in a lower FID and higher NDUS, indicating that the unlearning ability of Baseline-GA and SISS is partly due to the decline in generative performance. Appendix E shows more detailed results during unlearning Automobile including Baseine-F.

5.4 Experiments for Conditional DMs

We apply the DDPM with classifier-free guidance Ho & Salimans (2021) with a linear noise schedule and 1K diffusion steps. We randomly selected 1K images from each class of the CIFAR-10 dataset to construct D for the evaluation framework. We pretrain the model for 1M steps with the mini-batch size 128 on D and obtain the pre-trained DDPM $\boldsymbol{\theta}_p$, and then use it to generate 1K images of one class to construct $\hat{D}_p$. Then we select 16 images in D, which have the most duplicates in $\hat{D}_p$, to construct the unlearning set D_u and the remaining 9984 images of D construct the retain set D_r. We set $N_{\text{interval}} = 5$ and $\gamma = 0.1$.

Table 1: Performance Overview of the Proposed SMUD and Baselines on Unconditional DDPM

Unlearning Object	Method	FID_M (↓)	NDUS_M (↓)	FID_F (↓)	NDUS_F (↓)
Airplane	Baseline-GA	25.11	0	11.74	0
	SISS	91.43	12	11.36	22
	SMUD (ours)	12.42	0	11.37	0
Automobile	Baseline-GA	27.68	1	9.03	8
	SISS	86.78	3	10.39	25
	SMUD (ours)	9.02	0	8.61	0
Bird	Baseline-GA	22.30	2	13.17	1
	SISS	110.00	0	14.03	22
	SMUD (ours)	15.96	0	14.42	0
Cat	Baseline-GA	36.97	0	16.23	0
	SISS	155.99	0	16.04	36
	SMUD (ours)	16.14	0	15.71	0
Deer	Baseline-GA	26.32	4	11.18	4
	SISS	87.88	0	10.80	25
	SMUD (ours)	12.19	0	10.99	0

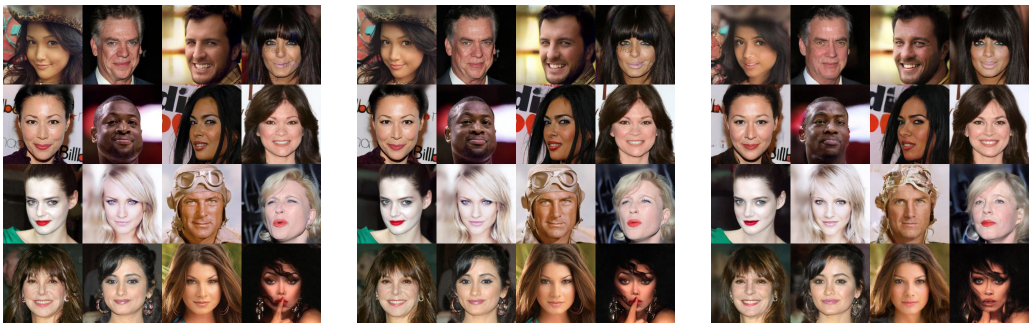
Table 2: Performance Overview of the Proposed SMUD and Baselines on Conditional DDPM

Unlearning Object	Method	FID_M(↓)	NDUS_M(↓)	FID_F(↓)	NDUS_F(↓)
Airplane	Baseline-GA	57.23	0	57.48	1
	SISS	134.06	0	56.02	40
	SMUD (Ours)	63.85	0	55.40	0
Automobile	Baseline-GA	39.00	0	28.69	20
	SISS	112.13	0	30.30	35
	SMUD (Ours)	45.80	0	29.10	0
Bird	Baseline-GA	80.85	1	44.58	15
	SISS	146.27	1	44.24	45
	SMUD (Ours)	54.56	0	42.41	0
Cat	Baseline-GA	69.25	0	54.56	53
	SISS	150.65	0	54.29	50
	SMUD (Ours)	63.03	0	51.99	0
Deer	Baseline-GA	54.34	0	49.75	10
	SISS	98.09	0	50.04	67
	SMUD (Ours)	60.85	0	46.49	0

Table 2 provides a performance overview of the proposed SMUD and two baselines across five CIFAR-10 classes. We apply SMUD and and Baseline-GA to unlearn the Automobile for 10K steps and each of the other four classes for 20K steps. The number of finetuning steps is set equal to the corresponding unlearning steps. For SISS, the unlearning is performed for approximately 50 steps, followed by 10K finetuning steps. As shown in Table 1, SISS leads to a substantial degradation in generative performance, while both SISS and Baseline-GA exhibit pronounced fake unlearning. SMUD maintains the best generative performance after unlearning and does not exhibit fake unlearning. Appendix F shows more detailed results during unlearning Automobile including Baseine-F.

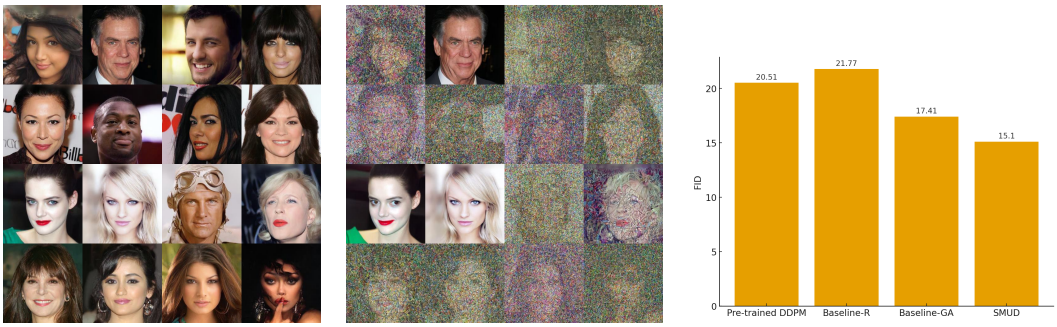
5.5 Experiments on CelebA-HQ Dataset

We use the CelebA-HQ dataset Odhiambo (2024), which contains 30K 256×256 images, to assess the proposed SMUD when the DM is trained on large datasets. We pre-train the unconditional DDPM on CelebA-HQ for 800K steps with the mini-batch size 32. We randomly select 128 images from CelebA-HQ to construct the unlearning set. The remaining 29,872 images construct the retain set. In this section, we present a qualitative evaluation as demonstrated in Fig. 1. We set $N_{\text{interval}} = 10$ and $\gamma = 1.0$. We find that the generative performance is significantly damaged after only 90 unlearning steps of SISS, as shown in Appendix D. Thus, we only qualitatively evaluate Baseline-R, Baseline-GA, and SMUD. The optimization steps for Baseline-GA and SMUD are 100K.



(a) Original unlearning images (b) Pre-trained DDPM (c) Baseline-R

Figure 3: 16 randomly selected unlearning set images and corresponding reconstructed images by the pre-trained DDPM and Baseline-R.



(a) Baseline-GA (b) SMUD (c) FID after unlearned

Figure 4: Reconstructed images by Baseline-GA and SMUD, and FID of DMs after being unlearned.

Figure 3a presents 16 images randomly selected from the unlearning set. Figures 3b and 3c show the corresponding reconstructions by the pre-trained DDPM and Baseline-R, respectively, where Baseline-R serves as the ideal unlearned DM. As illustrated in Fig. 3, the reconstructions generated by the pre-trained DDPM are nearly identical to the original images, while those from Baseline-R also exhibit high similarity. Figure 4 shows the reconstructed images generated by the DMs unlearned with Baseline-GA and SMUD. Comparing Figs. 4 and 3, the reconstructions from Baseline-GA visually resemble the original images more closely than those from Baseline-R. In contrast, the reconstructions produced by SMUD exhibit greater deviations, underscoring its effectiveness over Baseline-GA. Qualitative results of all 128 unlearning set images can be found in Appendix G. On the other hand, SMUD preserves the best generative performance as shown in Fig. 4c, suggesting that its superior unlearning performance is not due to any degradation in generative quality. Moreover, the DM exhibits higher generative performance after unlearning with SMUD compared to the pre-trained DM, indicating that SMUD minimally impacts generative quality.

6 Conclusion

In this paper, we first define two objectives for sample-level machine unlearning (SLMU) in diffusion models (DMs). We then propose a quantitative evaluation framework that leverages the memorization property of DMs to assess these objectives. This framework can be used to construct a benchmark for SLMU and thus lay a foundation for future research. Compared to the evaluation metrics proposed in the pioneering work on SLMU, our proposed evaluation framework provides a better assessment of unlearning methods and validates whether these methods can achieve complete unlearning. Additionally, we propose Sample-level Machine Unlearning for Diffusion models (SMUD), which modifies the generation path of DMs to prevent the generation of images in the unlearning set. Experimental results against baselines show that SMUD is the only method that does not exhibit fake unlearning in both unconditional and conditional DMs. Furthermore, SMUD preserves the highest generative performance after unlearning.

REFERENCES

Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj, 2016. Accessed: 2024-09-23.

California consumer privacy act of 2018. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375, 2018. Accessed: 2024-09-23.

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

Ricardo Baptista, Agnimitra Dasgupta, Nikola B Kovachki, Assad Oberai, and Andrew M Stuart. Memorization and regularization in generative diffusion models. *arXiv preprint arXiv:2501.15785*, 2025.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159. IEEE, 2021.

Jonathan Brophy and Daniel Lowd. Machine unlearning for random forests. In *International Conference on Machine Learning*, pp. 1092–1104. PMLR, 2021.

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.

Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. Recommendation unlearning. In *Proceedings of the ACM Web Conference 2022*, pp. 2768–2777, 2022a.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, pp. 499–513, 2022b.

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.

Tianqi Chen, Shujian Zhang, and Mingyuan Zhou. Score forgetting distillation: A swift, data-free method for machine unlearning in diffusion models. *arXiv preprint arXiv:2409.11219*, 2024a.

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.

Yunhao Chen, Xingjun Ma, Difan Zou, and Yu-Gang Jiang. Extracting training data from unconditional diffusion models. *arXiv preprint arXiv:2406.12752*, 2024b.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7210–7217, 2023.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gn0mIhQGNM.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12043–12051, 2024.

Masane Fuchi and Tomohiro Takagi. Erasing concepts from text-to-image diffusion models with few-shot unlearning. *arXiv preprint arXiv:2405.07288*, 2024.

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

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9304–9312, 2020a.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*, pp. 383–398. Springer, 2020b.

Aditya Golatkar, Alessandro Achille, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 792–801, 2021.

Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*, 2023.

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*, ICML'20. JMLR.org, 2020.

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. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

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

Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified gradient-based machine unlearning with remain geometry enhancement. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dheDf5EpBT.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36, 2024.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. *Improved precision and recall metric for assessing generative models*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Guihong Li, Hsiang Hsu, Chun-Fu Chen, and Radu Marculescu. Machine unlearning for image-to-image generative models. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=9hjVoPWPnh.

Xinfeng Li, Yuchen Yang, Jiangyi Deng, Chen Yan, Yanjiao Chen, Xiaoyu Ji, and Wenyuan Xu. SafeGen: Mitigating Sexually Explicit Content Generation in Text-to-Image Models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2024b.

Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, PRANAY SHARMA, Sijia Liu, et al. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10422–10431, 2022.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBsCjcPu_tE.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, pp. 931–962. PMLR, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Moses Odhiambo. Celeba-hq: Resized 256x256. https://www.kaggle.com/datasets/badasstechie/celebahq-resized-256x256, 2024. Accessed: 2024-10-21.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Javier Rando, Daniel Paleka, David Lindner, Lennart Heim, and Florian Tramer. Red-teaming the stable diffusion safety filter. In *NeurIPS ML Safety Workshop*, 2022. URL https://openreview.net/forum?id=zhDO3F35Uc.

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*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022a.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022b.

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.

Sebastian Schelter, Stefan Grafberger, and Ted Dunning. Hedgecut: Maintaining randomised trees for low-latency machine unlearning. In *Proceedings of the 2021 International Conference on Management of Data*, pp. 1545–1557, 2021.

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.

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*, 34:18075–18086, 2021.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023.

Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data–anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1451–1468, 2022.

Vinith Suriyakumar and Ashia C Wilson. Algorithms that approximate data removal: New results and limitations. *Advances in Neural Information Processing Systems*, 35:18892–18903, 2022.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 4007–4022, 2022.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a.

Wenhao Wang, Yifan Sun, Zongxin Yang, Zhengdong Hu, Zhentao Tan, and Yi Yang. Replication in visual diffusion models: A survey and outlook. *arXiv preprint arXiv:2408.00001*, 2024b.

Ga Wu, Masoud Hashemi, and Christopher Srinivasa. Puma: Performance unchanged model augmentation for training data removal. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8675–8682, 2022.

Jing Wu, Trung Le, Munawar Hayat, and Mehrtash Harandi. Erasediff: Erasing data influence in diffusion models. *arXiv preprint arXiv:2401.05779*, 2024a.

Yinjun Wu, Edgar Dobriban, and Susan Davidson. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning*, pp. 10355–10366. PMLR, 2020.

Yongliang Wu, Shiji Zhou, Mingzhuo Yang, Lianzhe Wang, Wenbo Zhu, Heng Chang, Xiao Zhou, and Xu Yang. Unlearning concepts in diffusion model via concept domain correction and concept preserving gradient. *arXiv preprint arXiv:2405.15304*, 2024b.

Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.

Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In *IJCAI*, volume 6, pp. 19, 2022.

TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

Xun Yuan, Yang Yang, Prosanta Gope, Aryan Pasikhani, and Biplab Sikdar. Vflgan: Vertical federated learning-based generative adversarial network for vertically partitioned data publication. In *Proceedings of Privacy Enhancing Technologies Symposium*, pp. 840–858, 2024a.

Xun Yuan, Zilong Zhao, Prosanta Gope, and Biplab Sikdar. Vflgan-ts: Vertical federated learning-based generative adversarial networks for publication of vertically partitioned time-series data. *arXiv preprint arXiv:2409.03612*, 2024b.

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

Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Forty-first International Conference on Machine Learning*, 2024b.

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*, 2024c.

A Related Work

In this section, we first review existing SLMU methods for classification tasks and discuss why most are inadequate for SLMU in DMs. Next, we review existing class-level MU methods for DMs. Last, we present the current evaluation frameworks for class-level MU methods in DMs, concluding that they are unsuitable for evaluating SLMU methods.

A.1 Sample-Level Machine Unlearning for Classification Models

Exact unlearning methods: Existing SLMU methods in classification models can be categorized into exact unlearning and approximate unlearning. Exact unlearning methods unlearn the specific data by retraining the model. In Bourtoule et al. (2021), the authors introduce the Sharding, Isolation, Slicing, and Aggregation (SISA) framework, a general approach for exact unlearning. SISA enables selective data removal by sharding, isolating, slicing, and aggregating training data, avoiding full model retraining. Building on this idea, Bourtoule et al. (2021); Yan et al. (2022); Chen et al. (2022a) utilize ensemble methods that split the dataset into smaller sub-datasets, training a separate model for each. The final classification is based on the combined outputs of these models. To lower computational costs, they delete the data to be unlearned from the relevant sub-datasets and retrain only the affected sub-models. Schelter et al. (2021); Brophy & Lowd (2021) apply the SISA framework to tree-based classification models, while Chen et al. (2022b) extends SISA for Graph Neural Networks (GNNs). However, the SISA framework is specifically designed for classification models that work with partitionable datasets, allowing the unlearning set to be isolated. In contrast, generative models operate by learning distributions rather than mapping inputs to outputs directly, as in classification tasks. The SISA framework, focusing on classification models, is less suitable for generative models since less data will lead to significant performance decreases, and the ensemble model in SISA cannot improve the generation capability compared to one DM trained on one small dataset.

Approximate unlearning methods: Approximate unlearning reduces the computational cost of exact unlearning and includes methods based on (1) influence function estimation, (2) model parameter re-optimization, and (3) gradient updates Xu et al. (2024). Influence function, introduced by Guo et al. (2020), estimates the influence of given training data using the first-order gradient and second-order gradient (Hessian matrix) according to the loss function of those data and is used to remove the data's influence on the model. Besides, random noise is added to the parameters's gradients during optimization to remove the influence completely. However, this method relies on convexity assumptions and involves costly Hessian matrix inversion, and random noise decreases the model's performance. Later research Sekhari et al. (2021); Suriyakumar & Wilson (2022); Mehta et al. (2022); Wu et al. (2022) developed more efficient approximations. Model re-optimization methods, like weight perturbation, partially retrain models to update parameters without full retraining Golatkar et al. (2020a;b; 2021), though they still involve Hessian approximations. However, in DMs, inputs combine images with random Gaussian noise, and the model learns to predict this noise. Since the noise varies with each training step, calculating the Hessian for a specific image and noise is ineffective for unlearning. The noise used for Hessian estimation differs from that used during training, making the Hessian irrelevant for capturing the influence of a particular training sample. Gradient-based unlearning methods generally follow two steps: (1) initialize model parameters from the previously trained model, and (2) apply a few gradient updates based on modified data. DeltaGrad Wu et al. (2020) adapts models efficiently to small training dataset changes by using cached gradients and parameter information. However, it is impractical due to the large memory required to store this information for every training iteration.

A.2 Machine Unlearning for Diffusion Model

DMs often utilize diverse open-source data, which can lead to the risk of incorporating sensitive or inappropriate information Chen et al. (2023b). This has raised concerns about the potential for generating harmful content Schramowski et al. (2023); Rando et al. (2022), violating copyright through the imitation of artistic styles Gandikota et al. (2023); Salman et al. (2023), or even memorizing training data Wang et al. (2024b); Somepalli et al. (2023); Carlini et al. (2023). Recent machine unlearning efforts in DMs focus on removing specific features or classes. For instance, Salun Fan et al. (2024) enables unlearning by fine-tuning only the most affected salient weights. Random labelling of the data from the unlearning set is used to update the salient weights. Forget-Me-Not Zhang et al. (2024a) leverages cross-attention scores to optimize the model's perception of target concepts. By optimizing vision-only self-attentive layers of stable diffusion using <nude, mosaic, benign> image triplets, Safegen Li et al. (2024b) remove pornographic latent representations from its attentive matrices, cutting off the associations between sexually-related text and nudity vision. In contrast, Fuchi & Takagi (2024) focuses on unlearning target concepts from the text encoder of Stable Diffusion via a gradient-ascent method without modifying the U-Net parameters. Erased Stable Diffusion

(ESD)Gandikota et al. (2023) fine-tunes model to align the conditional scores of undesired concepts with those of unconditioned, permanently removing learned concepts. Follows similar idea Chen et al. (2024a) aligns conditional scores of undesirable classes with those of safe classes. Wu et al. (2024b) frames the unlearning task as an adversarial training process, where the DM serves as the generator to predict noise, and a discriminator classifies whether the noise is linked to the target concept or the anchor concept. The objective is to align the DM's output between these two concepts. In Li et al. (2024a) and Wu et al. (2024a), the authors modify the reverse process of DMs by updating the loss function to align the predicted noise of specific concepts with a predefined noise distribution. However, the methods mentioned above are designed for conditional DMs to unlearn a class of data or feature and cannot solve the finer-grained sample-level machine unlearning. Sfront Huang et al. (2024) utilizes gradient ascent to achieve class-level machine unlearning. Although the method in Huang et al. (2024) is initially proposed for class-levl MU, it can be adapted to SLMU. In this paper, we employ Sfront as a baseline, termed as Baseline-GA, since it can be adapted to solve SLMU in DMs.

B Evaluation Framework

Overall evaluation framework. Algorithm 1 summarizes the overall process of the proposed evaluation framework. (i) Train a DM $\boldsymbol{\theta}_p$ on a dataset D as the pre-trained model. (ii) Generate a synthetic dataset $\hat{D}_p$ (the same size as D) with $\boldsymbol{\theta}_p$. (iii) Find memorized training images from $\hat{D}_p$ using the same method as Yoon et al. (2023); Gu et al. (2023). (iv) Select N_u most memorized training images to construct the unlearning set D_u and the retain set $D_r = D \setminus D_u$. (v) Apply an unlearning method $\mathcal{M}$ to obtain an unlearned DM $\boldsymbol{\theta}_u = \mathcal{M}(\boldsymbol{\theta}_p, D_r, D_u)$. (vi) Construct a synthetic dataset $\hat{D}_u$ (the same size as D) with $\boldsymbol{\theta}_u$. (vii) Select duplicates of D_u's data from $\hat{D}_u$ using the method in Yoon et al. (2023); Gu et al. (2023) and count the duplicates as NDUS. (viii) Fine-tune the unlearned DM $\boldsymbol{\theta}_u$ on the retain set and calculating NDUS of the fine-tuned DM, similar to steps (vi) and (vii). (ix) Last, the proposed evaluation framework returns NDUS after unlearning, NDUS after finetuning, FID between D_r and $\hat{D}_u$. The FID between $\hat{D}_u$ and D_r evaluates objective (i) of Definition 1. NDUSs of unlearned DM and fine-tuned DM evaluate objective (ii) of Definition 1.

Algorithm 1: Evaluation framework.

Input : Dataset D; initialized DM $\boldsymbol{\theta}_0$; training algorithm of DM $\mathcal{T}$; MU method $\mathcal{M}$.
Output: NDUS and FIDs

$\boldsymbol{\theta}_p \leftarrow \mathcal{T}(\boldsymbol{\theta}_0, D)$ // Train $\boldsymbol{\theta}_p$ on D according to **Eq.**(3);
$\hat{D}_p \leftarrow \text{DM}(\boldsymbol{\theta}_p, \mathbf{z_1}), \mathbf{z_1} \in \mathcal{N}(\mathbf{0}, \mathbf{I})$ // Generated $\hat{D}_p$ with $\boldsymbol{\theta}_p$;
Select duplicates of D's data from $\hat{D}_p$ using the method in Yoon et al. (2023); Gu et al. (2023)
Construct unlearning set D_u with N_u most memorized data
$D_r = D \setminus D_u$ // Construct retain set;
$\boldsymbol{\theta}_u = \mathcal{M}(\boldsymbol{\theta}_p, D_u, D_r)$ // Obtain unlearned model $\boldsymbol{\theta}_u$;
$\hat{D}_u \leftarrow \text{DM}(\theta_u, \mathbf{z_2}), \mathbf{z_2} \in \mathcal{N}$ // Generated $\hat{D}_u$ with $\boldsymbol{\theta}_u$;
Select duplicates of D_u's data from $\hat{D}_u$ using the method in Yoon et al. (2023); Gu et al. (2023) and count the duplicates as NDUS
$\boldsymbol{\theta}_f = \mathcal{T}(\boldsymbol{\theta}_u, D_r)$ // Finetune $\boldsymbol{\theta}_u$ on D_r;
$\hat{D}_f \leftarrow \text{DM}(\theta_u, \mathbf{z_3}), \mathbf{z_3} \in \mathcal{N}$ // Generated $\hat{D}_f$ with $\boldsymbol{\theta}_f$;
Compute NDUS by comparing $\hat{D}_f$ and D_u
Return NDUS after unlearning, NDUS after finetuning, FID$(D_r, \hat{D}_u)$

C Pseudo code of SMUD

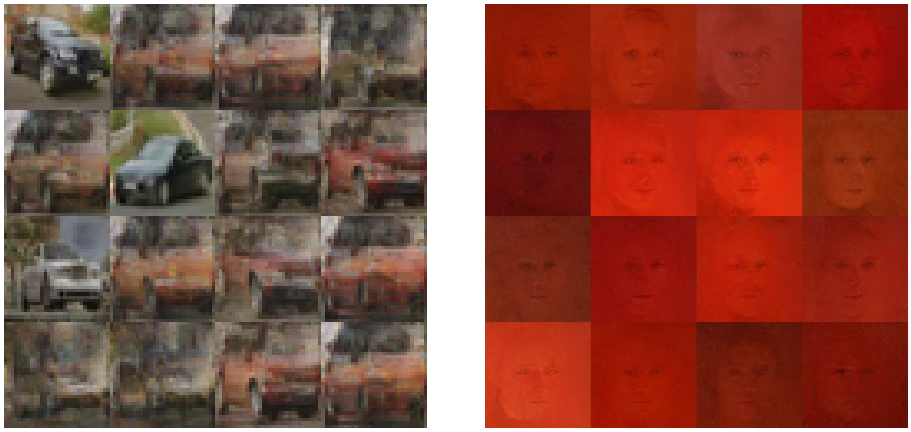
Algorithm 2 summarizes the proposed SMUD: (i) Sample images from the retain set and calculate the retain loss **Eq.**(9). (ii) If the optimization step n satisfies $n\%N_{\text{interval}} = 0$, copy the pre-trained model $\boldsymbol{\epsilon_\theta}$ into $\boldsymbol{\epsilon'_\theta}$ and sample images from the unlearning set. (iii) Calculate the unlearning loss **Eq.**(8) and add it to the retain loss. (iv) Optimize $\boldsymbol{\epsilon_\theta}$ by minimizing the final loss. (v) After $N_{unlearn}$ optimization steps, return the unlearned model.

Algorithm 2: Pseudo code of SMUD.

Input : Retain set D_r; Unlearning set D_u; pre-trained approximator $\boldsymbol{\epsilon_\theta}$; $\boldsymbol{\theta}_{\epsilon_\theta}$ denotes parameters of $\boldsymbol{\epsilon_\theta}$

Output: Unlearned approximator $\boldsymbol{\epsilon_\theta}$

for $n \in \{1, 2, \cdots, N_{unlearn}\}$ **do**
  Sample $\mathbf{x}_0^r$ from D_r
  // Retain loss **Eq.**(9) and **Eq.**(12);
  if *SMUD for unconditional DMs* **then**
    $l = \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon_\theta}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^r + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2$
  else if *SMUD for conditional DMs* **then**
    $l = \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon_\theta}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0^r + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{c}\right)\right\|^2$
  if $n\%N_{interval} = 0$ **then**
    $\boldsymbol{\epsilon}'_{\boldsymbol{\theta}} = \text{copy}(\boldsymbol{\epsilon_\theta})$
    Sample $\mathbf{x}_0^u$ from D_u
    // Unlearning loss **Eq.**(8) and **Eq.**(11);
    if *SMUD for unconditional DMs* **then**
      $\mathbf{y} = (\sqrt{\bar{\alpha}_t}\mathbf{x}_0^u + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)$
      $l = l + \|\boldsymbol{\epsilon}'_{\boldsymbol{\theta}}(\mathbf{y}) + \gamma\boldsymbol{\epsilon}' - \boldsymbol{\epsilon_\theta}(\mathbf{y})\|^2$
    else if *SMUD for conditional DMs* **then**
      $\mathbf{y_c} = (\sqrt{\bar{\alpha}_t}\mathbf{x}_0^u + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \mathbf{c})$
      $\mathbf{y}_\emptyset = (\sqrt{\bar{\alpha}_t}\mathbf{x}_0^u + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t, \emptyset)$
      $l = l + \|\boldsymbol{\epsilon}'_{\boldsymbol{\theta}}(\mathbf{y_c}) + \gamma\boldsymbol{\epsilon}' - \boldsymbol{\epsilon_\theta}(\mathbf{y_c})\|^2 + \|\boldsymbol{\epsilon}'_{\boldsymbol{\theta}}(\mathbf{y}_\emptyset) + \gamma\boldsymbol{\epsilon}' - \boldsymbol{\epsilon_\theta}(\mathbf{y}_\emptyset)\|^2$
  // Optimization step;
  $\boldsymbol{\theta}_{\epsilon_\theta} \leftarrow \boldsymbol{\theta}_{\epsilon_\theta} - \eta\nabla_{\boldsymbol{\theta}_{\epsilon_\theta}} l$
Return $\boldsymbol{\epsilon_\theta}$



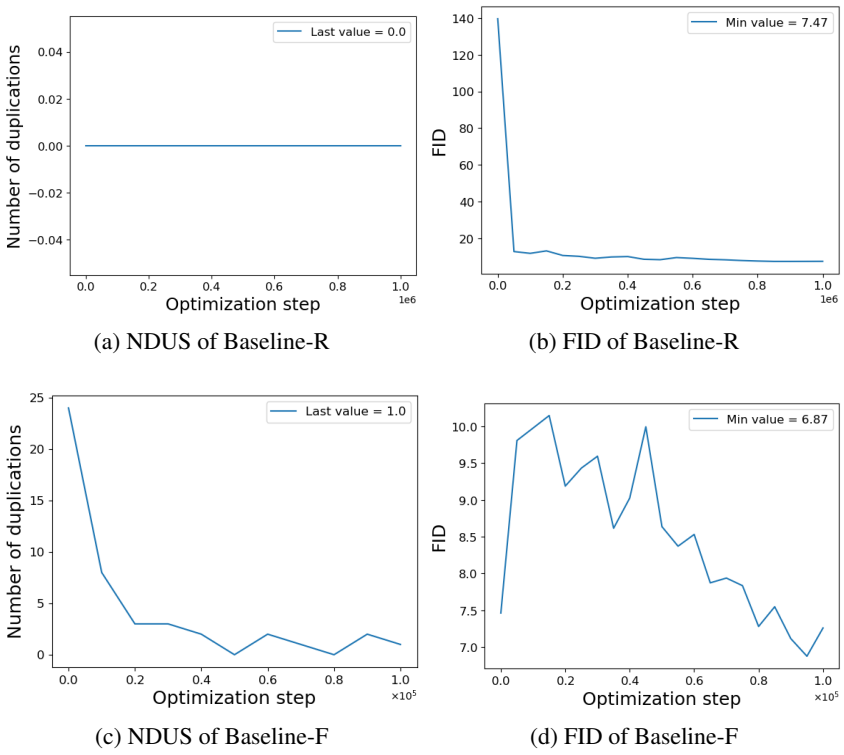
(a) 1.6K unlearning steps (CIFAR10) (b) 90 unlearning steps (celeb-HQ)

Figure 5: Synthetic images generated by unconditional DDPMs unlearned by SISS.

Figure 6: Synthetic images generated by conditional DDPMs unlearned by SISS for 70 steps.

D SUPPLEMENTAL RESULTS OF SISS

In this section, we explain why the unlearning process of SISS must be halted earlier. Figure 5a shows the synthetic images generated by the unconditional DDPM after being unlearned for 1.6K optimization steps with SISS on the CIFAR-10 dataset. Figure 5b shows the synthetic images generated by the unconditional DDPM after being unlearned for 90 optimization steps with SISS on the Celeb-HQ dataset. Besides, Fig. 6 shows the synthetic images generated by the conditional DDPM after being unlearned for 70 optimization steps with SISS on the CIFAR-10 dataset. As shown in the above synthetic images, the quality of the synthetic images is too poor, so the unlearning process needs to be stopped.



(a) NDUS of Baseline-R (b) FID of Baseline-R

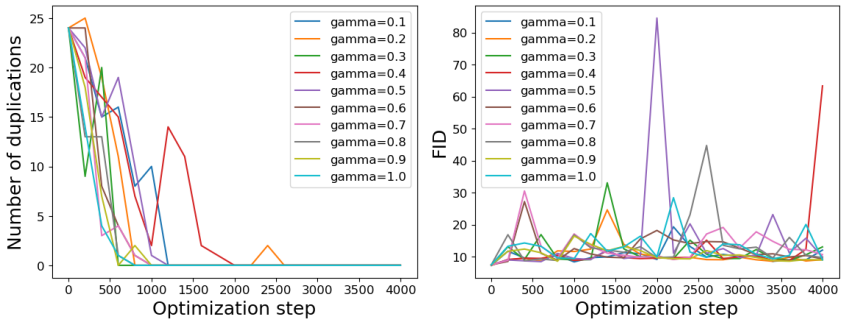
(c) NDUS of Baseline-F (d) FID of Baseline-F

Figure 7: NDUS and FID curves during the unlearning of unconditional DDPM using Baseline-R and Baseline-F.

E SUPPLEMENTAL RESULTS FOR UNCONDITIONAL DM

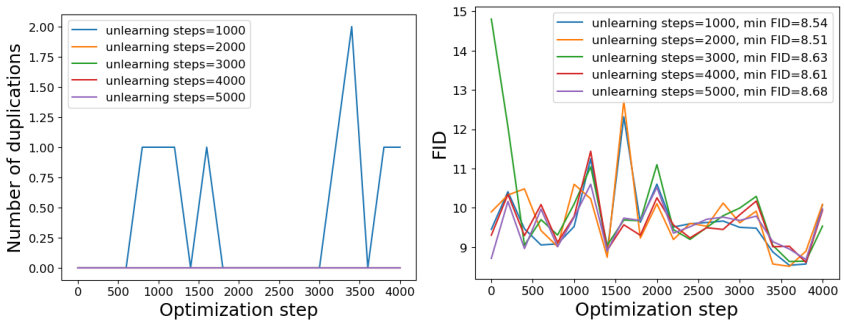
In this section, we demonstrate detailed results when unlearning Automobile. Figure 7 shows the NDUS and FID curves during the unlearning of unconditional DDPM using Baseline-R and Baseline-F. As shown in Fig. 7a, Baseline-R generates zero duplicates of the unlearning set images, retraining the DM from scratch on the retain set. Figure 7b shows that Baseline-R achieves a minimum FID score of 7.47 during 1M optimization steps. On the other hand, Baseline-F fails to unlearn within 100K optimization steps since it still generates a duplicate of an unlearning set image after 100K optimization steps, as shown in Fig. 7c. Figure 7d shows that fine-tuning the pre-trained DDPM on the retain set can decrease the FID score.

Figure 8 shows the NDUS and FID curves during the unlearning of unconditional DDPM using the proposed SMUD. Figure 8a shows the NDUS curves of various γ values over the optimization steps. As shown in Fig. 8a, the DDPM successfully unlearns the images from the unlearning set after 3K optimization steps for all γ values, and the value of γ has a limited influence on the unlearning speed. Figure 8b shows the FID curves over the optimization steps of various γ. As shown in Fig. 8b, FID scores of the DDPM fluctuate during the unlearning process and the value of γ has a limited influence on the amplitude of the fluctuation. The fluctuation in NDUS curves occurs because the DM has not fully unlearned the unlearning set. Consequently, the retain loss, which aids in improving generative performance, leads to increases in NDUS. The fluctuation in FID curves arises because the unlearning loss can reduce generative performance, which is why the retain loss is necessary. To evaluate the existence of fake unlearning, we finetune the unlearned DDPM with the retain loss on the retain set. Figure 9a shows the NDUS curves during the fine-tuning of the DDPMs, which have been unlearned by SMUD for $\{1\text{K}, 2\text{K}, 3\text{K}, 4\text{K}, 5\text{K}\}$ optimisation steps with $\gamma = 1.0$. The fake unlearning exists when the DDPM is unlearned for 1K optimization steps. After more unlearning optimization steps, the fake unlearning phenomenon disappears, which means a complete unlearning. On the other hand, fine-tuning the unlearned DDPM improves the DDPM's generative performance w.r.t. FID score, as shown in Fig. 9b.



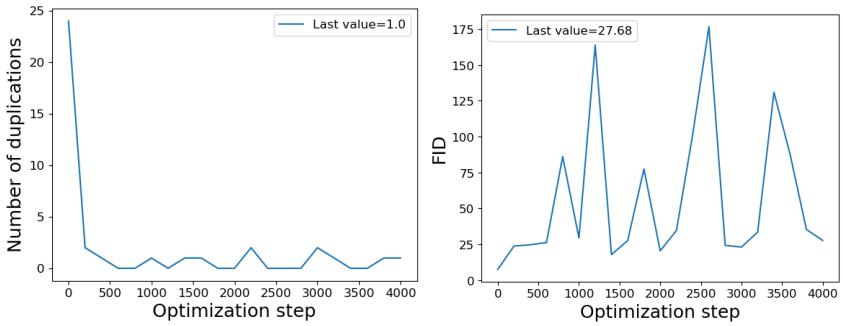
(a) NDUS curve (b) FID curve

Figure 8: NDUS and FID curves while unlearning the unconditional DDPM using SMUD.



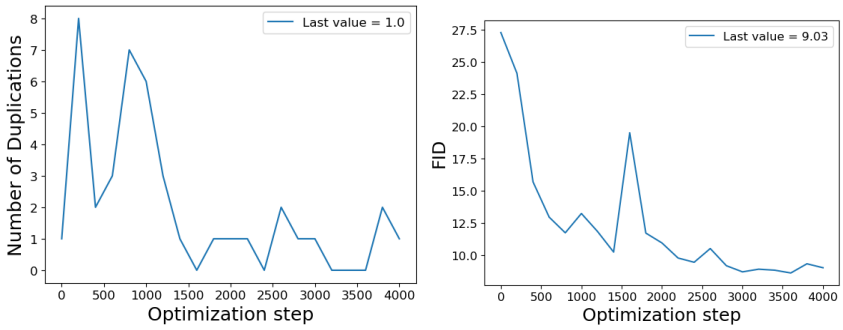
(a) NDUS curve (b) FID curve

Figure 9: NDUS and FID curves while fine-tuning the unlearned unconditional DDPM after being unlearned for 1K–5K optimization steps using SMUD with $\gamma = 1.0$.



(a) NDUS curve (b) FID curve

Figure 10: NDUS and FID curves while unlearning the unconditional DDPM using Baseline-GA.



(a) NDUS curve (b) FID curve

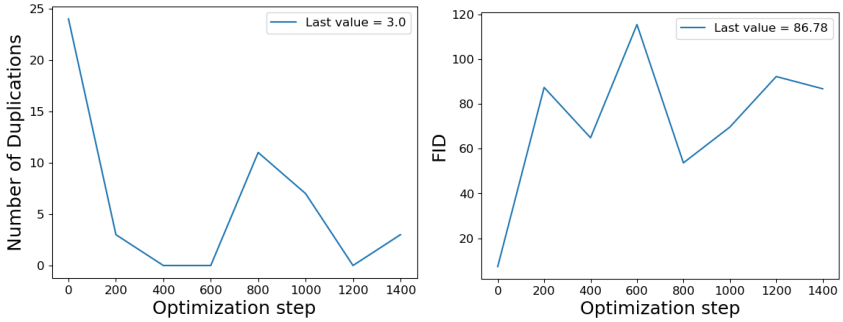
Figure 11: NDUS and FID curves while fine-tuning the unlearned unconditional DDPM after being unlearned for 4K optimization steps using Baseline-GA.

Figure 10 presents the NDUS and FID curves during unlearning unconditional DDPM using Baseline-GA. As shown in Fig. 10a, Baseline-GA fails to unlearn within 4K optimization steps since it still generates a duplicate of an unlearning set image after 4K optimization steps. Furthermore, as shown in Fig. 10b, the fluctuation in FID during unlearning with Baseline-GA is more pronounced than with SMUD, indicating that Baseline-GA degrades the performance of the DDPM to a greater extent. Moreover, as shown in Fig. 11, fine-tuning the DM unlearned by Baseline-GA can increase NDUS.

Figure 12 presents the NDUS and FID curves during the unlearning of the unconditional DDPM using SISS. As shown in Fig. 12b, SISS significantly reduces generative performance. We only plot the FID curve for up to 1400 unlearning steps, as the DM fails to generate any recognizable images beyond this point (refer to Appendix D for details). Furthermore, as shown in Fig. 12a, despite the significant decrease in generative performance, the DM still generates duplicates of the unlearning data even after 1400 unlearning steps. Furthermore, as illustrated in Fig. 13, fine-tuning the DM unlearned by SISS increases NDUS, suggesting that the unlearning ability of SISS is partly due to the decline in generative performance.

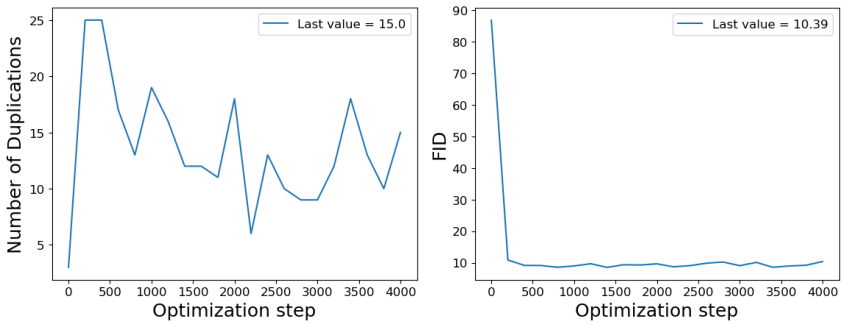
F Supplemental Results for Conditional DM

Figure 14 shows the NDUS and FID curves during unlearning conditional DDPM using Baseline-R and Baseline-F. As shown in Fig. 14a, Baseline-R generates zero duplicates of the unlearning set images retraining the DM from scratch on the retain set. Figure 14b shows that Baseline-R achieves a minimum FID score of 27.51 during 1M optimization steps. On the other hand, Baseline-F fails to unlearn the unlearning set within 100K optimization steps since it still generates 5 duplicates of unlearning set images after 100K optimization steps, as shown in Fig. 14c. Unlike the unconditional case, fine-tuning the pre-trained DDPM on the retain set cannot decrease the FID score for conditional DDPM, as shown in Fig. 14d. Besides, unlearning the conditional DDPM is more difficult



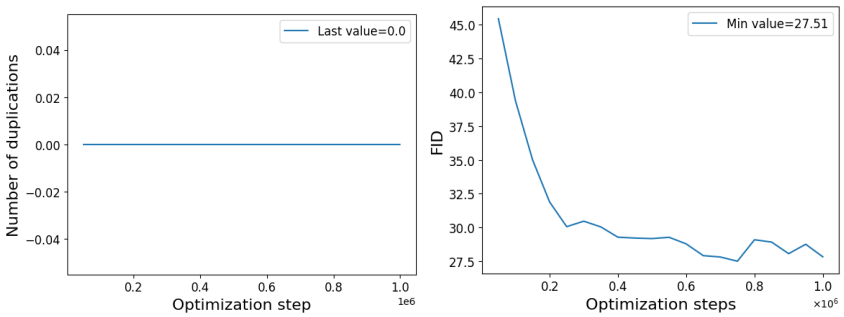
(a) NDUS curve (b) FID curve

Figure 12: NDUS and FID curves while unlearning the unconditional DDPM using SISS.

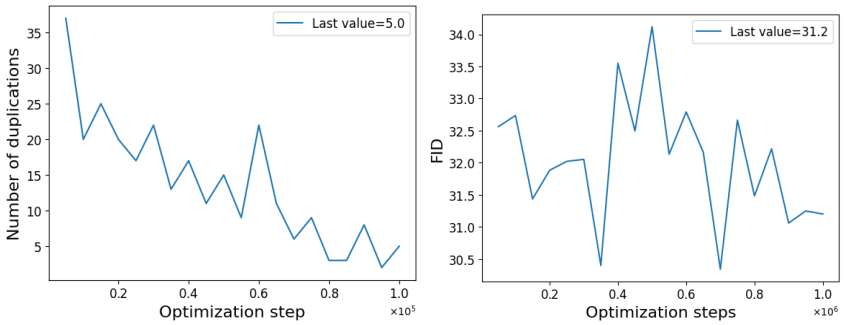


(a) NDUS curve (b) FID curve

Figure 13: NDUS and FID curves while fine-tuning the unlearned unconditional DDPM after being unlearned for 1.4K optimization steps using SISS.



(a) NDUS of Baseline-R (b) FID of Baseline-R



(c) NDUS of Baseline-F (d) FID of Baseline-F

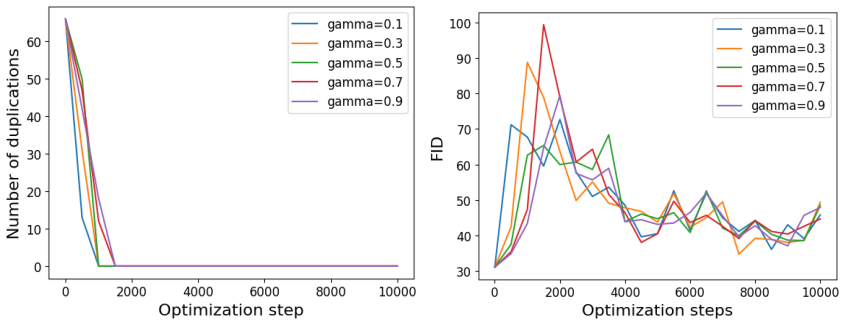
Figure 14: NDUS and FID curves during the unlearning of conditional DDPM using Baseline-R and Baseline-F.

than the unconditional DDPM, as evident from the comparison of Fig. 14c and Fig. 7c. Therefore, we perform 10K/20K optimization steps for unlearning in the conditional DDPM instead of 4K steps for the unconditional case.

Figure 15 shows the NDUS and FID curves during the unlearning of conditional DDPM using the proposed SMUD. Figure 15a shows the NDUS curves of various γ values over the optimization steps. As shown in Fig. 15a, the DDPM successfully unlearns the unlearning set images after 2K optimization steps for all γ values, and the value of γ has a limited influence on the unlearning speed. Figure 15b shows the FID curves over the optimization steps of various γ values. As shown in Fig. 15b, FID scores of the DDPM fluctuate during the unlearning process and the value of γ has a limited influence on the amplitude of the fluctuation. The fluctuation of the FID curves is because the unlearning loss can decrease the generative performance of the DM. To evaluate the existence of fake unlearning, we finetune the unlearned DDPM on the retain set. Figure 16a shows the NDUS curves during fine-tuning the DDPMs, which have been unlearned by SMUD for $\{2\text{K}, 4\text{K}, 6\text{K}, 8\text{K}, 10\text{K}\}$ optimisation steps with $\gamma = 0.1$. The fake unlearning phenomenon exists when the DDPM is unlearned for less than 8K optimization steps. After unlearning the DM over 8K optimization steps, the fake unlearning phenomenon disappears. On the other hand, fine-tuning after unlearning can help to improve the model's generative performance w.r.t. FID score, as shown in Fig. 16b.

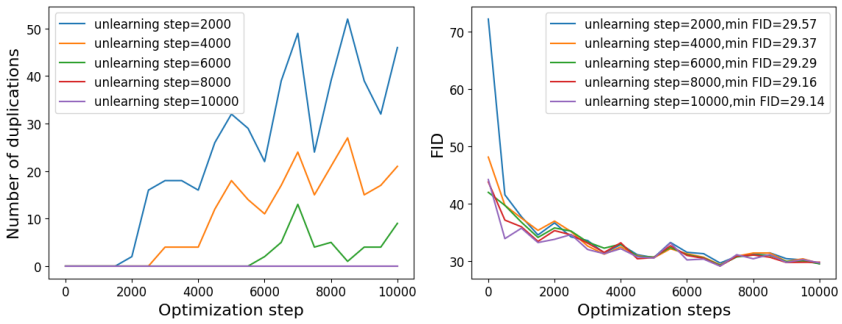
Figure 17 presents the NDUS and FID curves during unlearning conditional DDPM using Baseline-GA. Unlike the unconditional case, NDUS reaches 0 after 500 optimization steps, as shown in Fig. 17a. Then, we finetune the unlearned DDPM for 10K optimization steps, by minimizing the retain loss on the retain set. As shown in Fig. 18, NDUS increases significantly during the fine-tuning, which indicates that Baseline-GA did not achieve complete unlearning.

Figure 19 presents the NDUS and FID curves during unlearning conditional DDPM using SISS. Similar to the unconditional case, SISS significantly decrease the generative performance, as shown in Fig. 19b. We only plot the FID curve for up to 60 unlearning steps, as the DM fails to generate



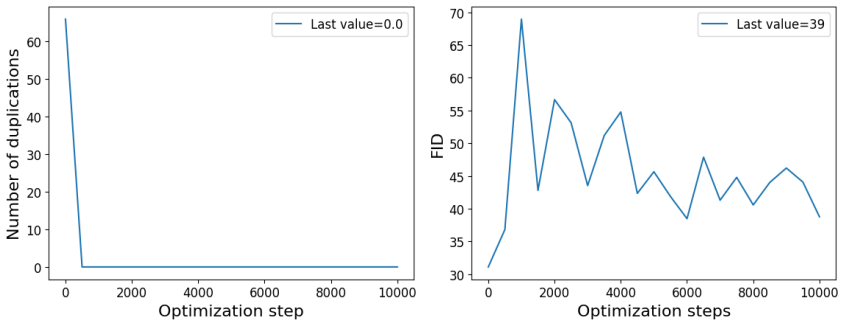
(a) NDUS curve (b) FID curve

Figure 15: NDUS and FID curves while unlearning the conditional DDPM using the proposed SMUD.



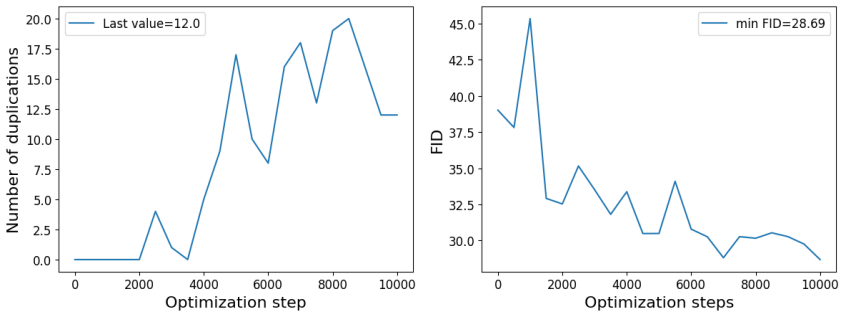
(a) NDUS curve (b) FID curve

Figure 16: NDUS and FID curves during fine-tuning the unlearned conditional DDPM after being unlearned for 2K–10K optimization steps using SMUD with $\gamma = 0.1$.



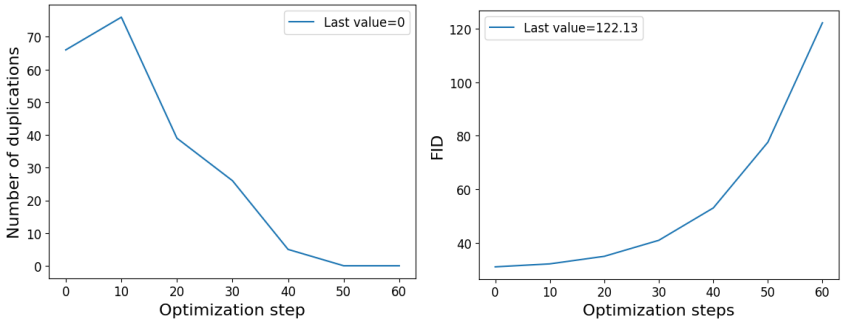
(a) NDUS curve (b) FID curve

Figure 17: NDUS and FID curves while unlearning the conditional DDPM using Baseline-GA.



(a) NDUS curve (b) FID curve

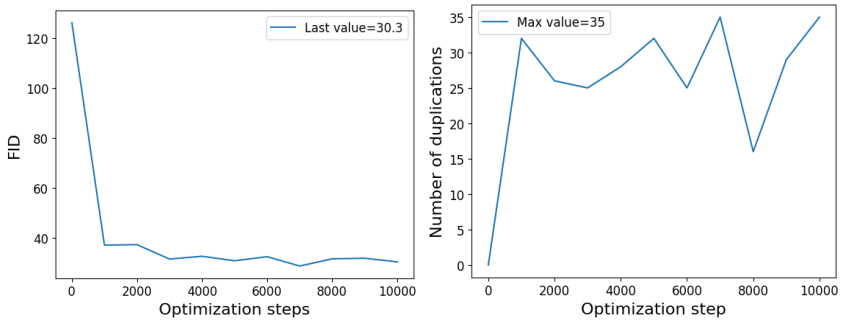
Figure 18: NDUS and FID curves while fine-tuning the unlearned conditional DDPM after being unlearned for 10K optimization steps using Baseline-GA.



(a) NDUS curve (b) FID curve

Figure 19: NDUS and FID curves while unlearning the conditional DDPM using SISS.

any recognizable images beyond this point (refer to Appendix D for details). Moreover, as shown in Fig. 20, fine-tuning the DM unlearned by SISS increases NDUS, indicating that SISS did not achieve complete unlearning.



(a) NDUS curve (b) FID curve

Figure 20: NDUS and FID curves while fine-tuning the unlearned conditional DDPM after being unlearned for 60 optimization steps using SISS.

G SUPPLEMENTAL RESULTS OF SMUD ON CELEBA-HQ

Following the qualitative evaluation framework introduced in Section 3.2, we reconstructed the unlearning set images from partially noised images after 400 forward steps. Figure 21 shows the unlearning set images, synthetic images reconstructed by the pre-trained DDPM and Baseline-R. The synthetic images reconstructed by the pre-trained DDPM closely resemble the corresponding unlearning set images, comparing Fig. 21a and Fig. 21b. Although Baseline-R is not trained on the unlearning set images, it can still reconstruct similar images because the partially noised images retain some information about the original images, as shown in Fig. 21c. However, the differences between Fig. 21c, and Fig. 21a are more pronounced compared to those between Fig. 21b and Fig. 21a if we zoom in.

Figure 22a and 22b show synthetic images reconstructed by the DDPM unlearned by Baseline-GA and SMUD, respectively, after 100K optimization steps. Consistent with Fig. 4, compared with the original unlearning images, the images reconstructed by SMUD show greater differences than Baseline-GA, highlighting its superiority over Baseline-GA.

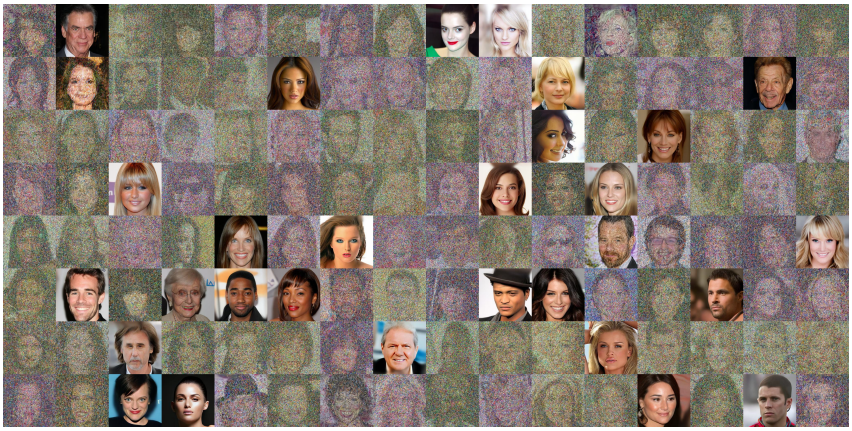
(a) Unlearning set images

(b) Reconstructed by the pre-trained DDPM

(c) Reconstructed by Baseline-R (trained on the retain set)

Figure 21: The unlearning set images and synthetic images reconstructed by the pre-trained DDPM and Baseline-R.

(a) Reconstructed by Baseline-GA



(b) Reconstructed by SMUD

Figure 22: Synthetic images reconstructed by the DDPMs unlearned with Baseline-GA and SMUD.