Efficient Unlearning through Maximizing Relearning Convergence Delay

Khoa Tran[1] Simon S. Woo[1,2*]
[1]CSE Department, Sungkyunkwan University, South Korea
[2]Secure Machines Lab
{khoa.tr, swoo}@g.skku.edu

Abstract

Machine unlearning poses challenges in removing mislabeled, contaminated, or problematic data from a pretrained model. Current unlearning approaches and evaluation metrics are solely focused on model predictions, which limits insight into the model's true underlying data characteristics. To address this issue, we introduce a new metric called relearning convergence delay, which captures both changes in weight space and prediction space, providing a more comprehensive assessment of the model's understanding of the forgotten dataset. This metric can be used to assess the risk of forgotten data being recovered from the unlearned model. Based on this, we propose the Influence Eliminating Unlearning framework, which removes the influence of the forgetting set by degrading its performance and incorporates weight decay and injecting noise into the model's weights, while maintaining accuracy on the retaining set. Extensive experiments show that our method outperforms existing metrics and our proposed relearning convergence delay metric, approaching ideal unlearning performance. We provide theoretical guarantees, including exponential convergence and upper bounds, as well as empirical evidence of strong retention and resistance to relearning in both classification and generative unlearning tasks.

1. Introduction

Deep learning has become the foundation of many commercial artificial intelligence (AI) systems, which provide users with powerful and convenient tools for everyday life. These models are trained on massive data, which includes both public datasets and private user data [41]. It inevitably raises serious concerns about privacy, trust, and user security. As the model's size and capacity increase, controlling its behaviors becomes more complex and challenging. The development of generative AI has led to the identification of new risks, including data poisoning, copyright infringement, disinformation, and the exploitation of personal identity. These risks may result in the unintentional disclosure of sensitive information, the leakage of private data, or the creation of biased, harmful, or copyright-infringing content by models. One of the most significant challenges for the future of responsible AI is striking a balance between the power of large, general-purpose models and the need for safer, more specialized systems.

To protect user rights, the European Union introduced the General Data Protection Regulation (GDPR) [42] and the Artificial Intelligence Act [10], which grant individuals the "right to be forgotten" [4], requiring technology companies to remove personal information from their databases and AI models upon request. As a result, machine unlearning [3, 15, 26, 37, 47], a technique aimed at selectively erasing specific learned knowledge or capabilities from AI models, has become increasingly essential. Beyond preserving privacy, machine unlearning plays a key role in ensuring fairness, accountability, and compliance with legal standards, in addition to safeguarding privacy.

In order to remove the impact of a particular subset of the dataset from a pretrained AI model, it is necessary to provide a model that is not capable of utilizing the characteristics of that subset, while at the same time preserving the advantageous behaviors that have been acquired from the retaining data [40]. Given the large size of the dataset, it is not feasible to retrain the model after removing the requested data, as this would require a significant amount of GPU usage and a prolonged training period. In addition, there is a possibility that private data will be unavailable at times, which will render retraining impossible. As a result, a few studies have proposed approximation unlearning [7, 9, 18, 24], which is not only feasible in terms of privacy access but also efficient in terms of time. Compared to the retraining method, the approximation unlearning approach involves beginning with a trained model and gradually modifying the weights of the model over a limited amount of time. This approach results in significantly lower costs being incurred compared to the retraining method.

The primary goals of machine unlearning are to achieve

*Corresponding Author

two primary objectives, which are utility and privacy guarantee [7]. The utility indicates that the model performs well on the retaining set. At the same time, privacy refers to exhibiting poor performance on the forgetting set and protecting the model against attacks on privacy, such as the leakage of private data. In image classification tasks, previous studies [7, 13, 35] employ accuracy to refer to the utility criteria and membership inference attacks (MIA) [5, 48] to refer to model leaking. In image generation tasks, machine unlearning aims to remove the model's ability to generate sensitive, harmful, or illegal content in response to inappropriate prompts [15, 18]. For evaluation, the Frechet Inception Distance (FID) score [27] is widely used and has demonstrated empirical consistency with human perceptual evaluations. However, FID, accuracy, and MIA focus on the model's predictions, ignoring the intermediate features that significantly contribute to those predictions and reveal the model's knowledge derived from the input data.

In this study, we propose a new metric and unlearning framework to improve the safety and effectiveness of machine unlearning. Our approach centers on measuring the influence of the forgetting data on the model through a novel metric and actively removing this influence using a principled strategy inspired from proposed metric. Specifically, we quantify the influence level of forgetting dataset via *relearning convergence delay*, and design a *Influence Eliminating Unlearning framework* to make the unlearning process more efficient and resistant to relearning.

We summarize our key contributions as follows:

- ***Relearning convergence delay* metric:** We introduce *relearning convergence delay* as a novel metric to quantify how quickly a model relearns forgotten data. Whereas existing relearning-time metrics quantify relearning in seconds [19, 49] and offer no explicit guidance for improvement, our metric reframes the problem in terms of convergence properties, providing a principled and explicit direction for reducing relearning risk.
- ***Influence Eliminating Unlearning* (IEU) framework:** We develop the **IEU** framework, which integrates *Gradient Ascent* to reverse the effect of the forgetting data and *Noisy Regularization* to delay the recovery risk of forgotten information, while preserving accuracy on the retaining set. Experiments show that **IEU** outperforms existing methods across existing metrics and our proposed metric in both classification and generation tasks.
- **Theoretical Guarantees:** We present a theoretical analysis that establishes the upper bound of our *relearning convergence delay* metric and develop an efficient approximation to make its computation practical. In addition, we derive an upper bound on the error of our **IEU** framework, offering clearer insight into the distinct contributions and interactions of each component within the framework.

2. Background

We denote $f(x, \theta)$ as a model parameterized by trainable weight θ, a training dataset $\mathcal{D}^{\text{train}} = \{x_i, y_i\}^N$ where x_i represents an input and y_i is a corresponding label, a testing dataset $\mathcal{D}^{\text{test}}$, and a training algorithm $\mathcal{T}(\theta_0, \mathcal{D}, t)$ [31] (such as Gradient Descent, Adam etc.) [32]. In the training process, the algorithm $\mathcal{T}$ tries to minimize the loss function on the training dataset $\mathcal{L}(\theta, \mathcal{D}^{\text{train}})$. We define a model is well-trained on $\mathcal{D}$ if $\theta^{\mathcal{D}} = \arg\min_\theta \mathcal{L}(\theta, \mathcal{D})$. In general, we define $\Phi(\theta, \mathcal{D})$ as an error-evaluation function for the model $f(\cdot, \theta)$ on the dataset $\mathcal{D}$. The function Φ could be a loss function $\mathcal{L}$ or the accuracy error $1 - \textit{accuracy}$ in the context of classification problems.

Machine Unlearning. In the unlearning scenario, the model is trained first on the entire training set and obtains an optimal weight $\theta^{\mathcal{D}^{\text{train}}} = \mathcal{T}(\theta_0, \mathcal{D}^{\text{train}}, +\infty)$ where θ_0 is an initialized weight. In the unlearning phase, we aim to reduce the influence of a part $\mathcal{D}_f^{\text{train}} \subset \mathcal{D}^{\text{train}}$, called forgetting set, from the trained weight $\theta^{\mathcal{D}^{\text{train}}}$. Ideally, we should retrain an initialized weight on the retaining set $\mathcal{D}_r^{\text{train}} = \mathcal{D}^{\text{train}} \setminus \mathcal{D}_f^{\text{train}}$, referred to as *exact unlearning*. However, it is infeasible due to time complexity, hardware cost, or privacy restrictions. Thus, *approximation unlearning* is proposed to make unlearning process fast and efficient by fine-tuning from a trained weight $\theta_0^{UL} = \theta^{\mathcal{D}^{\text{train}}}$ using an unlearning process $\theta_T^{UL} = \mathcal{U}(\theta_0^{UL}, \mathcal{D}_r, \mathcal{D}_f, T)$ in T iterations. The goal of approximation unlearning is to produce an unlearned model that performs similarly to the exact unlearning model. In the scenario of unlearning for a publicly pretrained model where the retaining dataset is inaccessible, the unlearning process becomes more challenging. In this case, the model update is defined as $\theta_T^{UL} = \mathcal{U}(\theta_0^{UL}, \emptyset, \mathcal{D}_f, T)$ over T steps, where no retaining data is used.

Unlearning Metrics. In classification tasks, previous studies [7, 13, 19, 35] commonly employ accuracy as a means of evaluating the performance of unlearned models, utilizing the corresponding retraining model as a benchmark. The underlying assumption is that a properly unlearned model should closely match the retraining model in behavior, reflected in comparable accuracy across the retaining, forgetting, and testing datasets.

In image generation tasks, the FID measures how closely the distribution of generated images aligns with that of real images, and it is the current standard for evaluating the quality of generative models [13, 28, 53]. A lower FID score indicates more realistically generated images, reflecting the effectiveness of the generative model. In the context of unlearning NSFW (not safe for work) content, previous works [13, 53] evaluate a model's ability to generate harmful content by employing detection models that assess the level of harmfulness in the generated images.

To assess privacy guarantees, previous works [13, 35]

employ MIA [48], which uses the outputs of the unlearned model to measure the attack success rate (ASR). MIA aims to determine whether a specific data sample was part of the model's training set, regarding the risk of information leaking, and ASR is widely used to assess the effectiveness of privacy in unlearning methods. Ideally, a practical forgetting method should achieve an MIA score comparable to that of a model retrained without the forgotten data.

Beyond MIA, the relearning attack [11, 14, 22, 30, 39] poses a privacy threat due to the model's long lifetime, as the unlearned model could potentially reacquire previously forgotten data, thereby challenging its robustness and weakening the guarantees of unlearning. Prior studies [19, 49] have assessed unlearning effectiveness by measuring the time (in seconds) required to relearn forgotten data. However, this metric offers limited insight into the learning dynamics and provides little guidance for improving unlearning methods. To the best of our knowledge, this is the first study to introduce a metric, *relearning convergence delay*, that quantifies the risk of forgotten data recovery in terms of convergence behavior. This addresses a critical gap in current unlearning evaluation practices by offering a more principled and informative measure of residual influence.

3. Relearning Convergence Delay Metric

How can we quantify the contribution of a dataset $\mathcal{D}$ to a learned model θ? Transfer learning offers a useful perspective: models pretrained on relevant data tend to converge faster on downstream tasks than those initialized randomly, even if their initial accuracies are similar. This implies that pretrained weights encode latent knowledge beneficial for learning, which is not always evident in performance metrics but is observable through training efficiency. Building on this insight, we hypothesize that the influence of dataset $\mathcal{D}$ on model weight θ can be quantified by the model's convergence speed during fine-tuning $\mathcal{T}(\theta, \mathcal{D}, \cdot)$.

In the context of unlearning, this has significant privacy implications. A model that retains significant influence from forgotten data may relearn it quickly, a vulnerability exploited by relearning attacks. To capture this, we propose a novel metric called *relearning convergence delay* ($\mathcal{RCD}$), which quantifies the residual influence of a forgetting set on an unlearned model. Specifically, $\mathcal{RCD}$ measures how efficiently an unlearned model θ_T^{UL} relearns on the forgotten dataset $\mathcal{D}_f$, thus serving as a proxy for the model's susceptibility to relearning attacks. It is formally defined as:

$$\mathcal{RCD}_{\mathcal{T}}(\theta_T^{UL}, \mathcal{D}_f) = \int_0^{+\infty} \left[\Phi(\mathcal{T}(\theta_T^{UL}, \mathcal{D}_f, t), \mathcal{D}_f) - \Phi(\theta^{\mathcal{D}_f}, \mathcal{D}_f) \right] dt, \tag{1}$$

where $\mathcal{T}$ is a learning algorithm. In the unlearning process, the objective is to eliminate the influence of the forgetting set on the model. To reflect this, we seek to maximize the *relearning convergence delay*, such that the unlearned model requires significantly more effort to relearn the forgotten data, indicating effective removal of its influence. To facilitate theoretical analysis and ensure convergence, we assume that the training algorithm $\mathcal{T}$ can achieve optimal model parameters under standard conditions.

Assumption 1. *The training algorithm $\mathcal{T}$ converges to the optimal parameter at the end of the training process, denoting as $\mathcal{T}(\theta, \mathcal{D}, +\infty) = \theta^{\mathcal{D}}$ for every θ and $\mathcal{D}$.*

To control the *relearning convergence delay* $\mathcal{RCD}$, we investigate the condition number [55], which is well known for representing the difficulty of convergence in a convex optimization problem. The investigation focuses on the convergence characteristics of iterative optimization algorithms. We denote that the loss function $\mathcal{L}(\theta, \mathcal{D})$ at θ on the dataset $\mathcal{D}$ has a second-order derivative $\nabla^2\mathcal{L}(\theta, \mathcal{D})$ which contains eigenvalues represented by the notation $\lambda_1(\theta, \mathcal{D}) \geq \lambda_2(\theta, \mathcal{D}) \geq \ldots \geq \lambda_d(\theta, \mathcal{D}) \geq 0$. The work [57] demonstrated that the condition number is minimized during the training process. Leveraging on this phenomenon, we are going to make the following assumption:

Assumption 2. *For every iterative and convertible learning algorithm $\mathcal{T}$, dataset $\mathcal{D}$, and initialized weight $\theta_0 \in \mathbb{R}^d$, the training process $\theta_t = \mathcal{T}(\theta_0, \mathcal{D}, t)$ progressively minimizes the condition number over time:*

$$\frac{\lambda_1(\theta_0, \mathcal{D})}{\lambda_d(\theta_0, \mathcal{D})} \geq \frac{\lambda_1(\theta_1, \mathcal{D})}{\lambda_d(\theta_1, \mathcal{D})} \geq \ldots \geq 1.$$

Lemma 3. *For Φ is a $\mu-$strongly and $\beta-$smooth loss function, every iterative and convertible learning algorithm $\mathcal{T}$, dataset $\mathcal{D}$, initialized weight $\theta_0 \in \mathbb{R}^d$, and training process $\theta_t = \mathcal{T}(\theta_0, \mathcal{D}, t)$, we have these properties:*
(a) $0 \leq \mu \leq \min_t \lambda_d(\theta_t, \mathcal{D})$
(b) $\beta \geq \max_t \lambda_1(\theta_t, \mathcal{D}) \geq 0$
(c) $\frac{\beta}{\mu} \geq \frac{\lambda_1(\theta_0, \mathcal{D})}{\lambda_d(\theta_0, \mathcal{D})} \geq \frac{\lambda_1(\theta_1, \mathcal{D})}{\lambda_d(\theta_1, \mathcal{D})} \geq \ldots \geq 1.$

Consequently, based on Lemma 3, it can be inferred that all eigenvalues and the condition number during the training process are bounded by a well-known assumption regarding strongly and smoothly convex loss functions.

While $\mathcal{RCD}_{\mathcal{T}}$ depends on the choice of the learning algorithm $\mathcal{T}$, in this paper, we derive its bound under a specific configuration where $\mathcal{T}$ is set to Gradient Descent. Building on the iterative update rule $\theta_{t+1} = \theta_t - \eta_t \nabla_t$, we make an analysis of $\mathcal{RCD}_{GD}$ bounds:

Theorem 4. *For Φ is a convex loss function, $\mathcal{T}$ is the Gradient Descent with step-size $\eta_t = \frac{1}{\lambda_1(\theta_t, \mathcal{D}_f)}$, the $\mathcal{RCD}_{GD}$ value is bounded by:*

$$0 \leq \mathcal{RCD}_{GD}$$

$$\leq \frac{\lambda_1(\theta_T^{UL}, \mathcal{D}_f)}{\lambda_d(\theta_T^{UL}, \mathcal{D}_f)} \Big(\mathcal{L}(\theta_T^{UL}, \mathcal{D}_f) - \mathcal{L}(\theta^{\mathcal{D}_f}, \mathcal{D}_f) \Big). \quad (2)$$

Theorem 4 implies that the $\mathcal{RCD}_{GD}$ is consistently non-negative and possesses an upper limit. The upper limit of $\mathcal{RCD}_{GD}$ for the weight θ_T^{UL} is dependent upon the condition number of the unlearned weight on the forgetting dataset $\frac{\lambda_1(\theta_T^{UL}, \mathcal{D}_f)}{\lambda_d(\theta_T^{UL}, \mathcal{D}_f)}$ and the loss function on forgetting set $\mathcal{L}(\theta_T^{UL}, \mathcal{D}_f)$, where the value of $\mathcal{L}(\theta^{\mathcal{D}_f}, \mathcal{D}_f)$ is independent of the unlearned weight. The condition number represents the difficulty of re-learning forgotten information, whereas the loss function value of the forgetting set reflects the performance of the unlearned model on the forgetting dataset. In other words, the upper bound of $\mathcal{RCD}_{GD}$ represents the worst case of relearning attack, which measures the cost required to ensure the success of relearning attack.

In general, we establish Corollary 5, which indicates that the *relearning convergence delay* $\mathcal{RCD}_{GD}$ is non-negative and bounded when the training algorithm $\mathcal{T}$ is Gradient Descent, for any unlearned model weight θ and dataset $\mathcal{D}$, assuming the loss function is μ-strongly convex and β-smooth. This result suggests that in general $\mathcal{RCD}_{GD}$ reflects both the model's current performance and the optimization difficulty on the forgotten dataset.

Corollary 5. *For Φ is a $\mu-$strongly and $\beta-$smooth convex loss function, $\mathcal{T}$ is the Gradient Descent with step-size $\eta_t = \frac{1}{\lambda_1(\theta_t, \mathcal{D})}$, for any θ and $\mathcal{D}$, the $\mathcal{RCD}_{GD}$ is bounded by:*

$$0 \leq \mathcal{RCD}_{GD}(\theta, \mathcal{D}) \leq \frac{\beta}{\mu} \Big(\mathcal{L}(\theta, \mathcal{D}) - \mathcal{L}(\theta^{\mathcal{D}}, \mathcal{D}) \Big). \quad (3)$$

While $\mathcal{RCD}$ is defined as an infinite integral, which is not feasible to compute in practice, we approximate it using a discrete and finite number of K iterations:

$$\mathcal{RCD}_{\mathcal{T}}^{K}(\theta_{UL}, \mathcal{D}_f) = \sum_{t=0}^{K} \Big[\Phi(\mathcal{T}(\theta_T^{UL}, \mathcal{D}_f, t), \mathcal{D}_f) - \Phi(\theta^{\mathcal{D}_f}, \mathcal{D}_f) \Big]. \quad (4)$$

Theorem 6. *By approximating the relearning convergence delay from Eq. (1) using Eq. (4), with $\mathcal{T}$ set to Gradient Descent, we obtain the following approximation error:*

$$\mathcal{RCD}_{GD} - \mathcal{RCD}_{GD}^{K} \leq \mathcal{O}(e^{-K}). \quad (5)$$

We introduce Theorem 6, which concerns the estimation error of approximated $\mathcal{RCD}_{GD}$ from the Eq. (4). This theory indicates the trade-off between the number of iterations and the precision of the approximation; for more iterations, we achieve a more accurate estimation of the *relearning convergence delay*. Significantly, it claims exponential convergence, indicating that a sufficient number of iterations can precisely yield an estimated score.

Algorithm 1 Influence Eliminating Unlearning framework

Input: weight $\theta^{\mathcal{D}} \in \mathbb{R}^d$, retaining data $\mathcal{D}_r$, forgetting data $\mathcal{D}_f$, noisy ratio $\alpha \in [0, 1]$, step-size $\eta > 0$, and forgetting set weight $c \in [0, 1]$
for $t = 1$ **to** T **do**
  Draw $\theta_{init} \overset{\text{iid}}{\sim} \mathcal{N}(0, \frac{2}{d})$
  Calculate ∇_{t-1}^{r} regarding loss function in Eq. (6) on retaining set
  Calculate ∇_{t-1}^{f} regarding loss function in Eq. (7) on forgetting set
  $\theta_t = \alpha\theta_{t-1} + (1-\alpha)\theta_{init} - \eta\nabla_{t-1}^{r} + c\eta\nabla_{t-1}^{f}$
end for
Return: θ_T.

4. Influence Eliminating Unlearning

The goal of the unlearning process is to remove the influence of the forgetting dataset while preserving performance on the retaining set. To ensure utility, we first apply a loss function to the retaining data, guiding the model to maintain its original performance. To mitigate the impact of the forgetting set, we introduce two key components, *Gradient Ascent* and *Noisy Regularization*, inspired by maximizing the *relearning convergence delay* score, thereby effectively minimizing the impact of data being forgotten on the original model. Our overall approach is formalized in Algorithm 1, named the *Influence Eliminating Unlearning* framework, which contains several hyperparameters related to the learning rate, forgetting rate, and noisy factor. For notation, the gradients at step t for the retaining and forgetting sets are denoted as ∇_t^r and ∇_t^f, respectively, and their second-order derivatives are $\nabla_t^{r\,2}$ and $\nabla_t^{f\,2}$.

4.1. Maintain the Performance on the Retaining Set

Fine-tuning a model on new data leads to catastrophic forgetting [1, 17, 51], where performance on previously learned data decreases. Without access to the retaining set, it becomes difficult to preserve its accuracy during model updates, resulting in degraded utility, contrary to the goal of unlearning. To address this, we first employ minimizing a loss function on the retaining set to maintain its performance and ensure the model's utility:

$$\mathcal{L}(\theta, \mathcal{D}_r^{\text{train}}) = \frac{1}{|\mathcal{D}_r^{\text{train}}|} \Sigma_i \ell(f(\theta, x_i), y_i). \quad (6)$$

4.2. Eliminate the Influence of Forgetting Set

Inspired by Theorem 4 and Corollary 5, we aim to maximize $\mathcal{RCD}$, which entails reducing the influence of the forgetting set $\mathcal{D}_f$ on the unlearned model θ_T^{UL}. As discussed in the previous section, it contains two factors: the loss function value $\mathcal{L}(\theta_T^{UL}, \mathcal{D}_f)$ and the condition number

$\frac{\lambda_1(\theta_T^{UL}, \mathcal{D}_f)}{\lambda_d(\theta_T^{UL}, \mathcal{D}_f)}$. We will discuss each one in this section.

4.2.1. Gradient Ascent.

We aim to degrade the model's performance on the forgetting set, ensuring a high loss for those data points. While catastrophic forgetting can be indirectly leveraged by minimizing loss only on the retaining set, causing the forgetting set's performance to decline over time, we explicitly maximize the loss on the forgetting set during unlearning. This targeted approach is expected to ensure the model forgets the specified data more effectively:

$$\mathcal{L}(\theta, \mathcal{D}_f^{\text{train}}) = \frac{1}{|\mathcal{D}_f^{\text{train}}|} \Sigma_i \ell(f(\theta, x_i), y_i). \tag{7}$$

In our ablation experiments, we empirically validate the effectiveness of utilizing gradient ascent on the forgetting set. While gradient ascent has the potential to introduce instability during training, we control and mitigate this risk by using a smaller step-size for the forgetting set updates, defined as $\eta_f = c\eta_r, c \in [0, 1)$, where η_r is the step-size for gradient descent on the retaining set.

4.2.2. Noisy Regularization.

The condition number of a neural network's weight is typically higher at initialization [6, 23, 34, 54] than after training, and it tends to decrease progressively throughout the training process [57]. We first assume that the weights are initialized using Kaiming normal initialization.

Assumption 7. *The initialized weight θ_0 follows Kaiming initialization $\mathcal{N}(0, \frac{2}{d})$ where $\theta \in \mathbb{R}^d$.*

We aim to design an iterative process that maximizes the condition number of the model weight θ on the dataset $\mathcal{D}$, given a well-trained model $\theta^{\mathcal{D}}$. Inspired by Assumption 7, we define the Iterative Re-initialization Process in Definition 8 by weighted merging the current weight and a randomly initialized weight, controlled by a parameter $\alpha \in [0, 1]$. In other words, it is an incorporation of weight decay and noisy injection in the weight space.

Definition 8. *The Iterative Re-initialization Process:*

$$\theta_{t+1} = \alpha\theta_t + (1 - \alpha)\mathcal{N}(0, \frac{2}{d}), \tag{8}$$

where $\alpha \in [0, 1]$ denotes the speed of process.

The model's weight, when applied to the Iterative Re-initialization Process with a sufficient number of iterations, will conform to a normal distribution in Assumption 7 and may be regarded as an initialized weight. A smaller α signifies a rapid process, whereas a larger α denotes a slow process. While a learning algorithm attempts to process an initialized weight into an optimal weight, represented as $\theta_0 \to \theta^{\mathcal{D}}$, the process described in Definition 8 executes the inverse function $\theta^{\mathcal{D}} \to \theta_0$. Therefore, according to Lemma 3, we can say that the Iterative Re-initialization Process maximizes the number of expectation conditions in the data set $\mathcal{D}$; however, it is still restricted to the setting of μ-strongly and β-smoothly convex, presented in Lemma 9.

Lemma 9. *The Iterative Re-initialization Process maximizes the condition number over the dataset $\mathcal{D}$ for $\theta_0 = \theta^{\mathcal{D}}$:*

$$1 \leq \mathbb{E}\left[\frac{\lambda_1(\theta_0, \mathcal{D})}{\lambda_d(\theta_0, \mathcal{D})}\right] \leq \mathbb{E}\left[\frac{\lambda_1(\theta_1, \mathcal{D})}{\lambda_d(\theta_1, \mathcal{D})}\right] \leq \ldots \leq \frac{\beta}{\mu}. \tag{9}$$

Finally, we summarize our proposed unlearning framework in Algorithm 1, which consists of three key components corresponding to three objectives: (a) minimizing the loss on the retaining set $\mathcal{L}(\theta, \mathcal{D}_r^{\text{train}})$, (b) maximizing the loss on the forgetting $\mathcal{L}(\theta, \mathcal{D}_f^{\text{train}})$, and (c) applying the Iterative Re-initialization Process (Definition 8) to eliminate the influence of the forgetting set. Specifically, component (a) seeks to reduce the loss on $\mathcal{D}_r$ and decrease the condition number ratio on the retaining set $\mathbb{E}\left[\frac{\lambda_1(\theta_{t+1}, \mathcal{D}_r)}{\lambda_d(\theta_{t+1}, \mathcal{D}_r)}\right] \leq \mathbb{E}\left[\frac{\lambda_1(\theta_t, \mathcal{D}_r)}{\lambda_d(\theta_t, \mathcal{D}_r)}\right]$, while component (b) aims to increase the loss on $\mathcal{D}_f$, and component (c) targets increasing the condition number ratio on the forgetting set $\mathbb{E}\left[\frac{\lambda_1(\theta_{t+1}, \mathcal{D}_f)}{\lambda_d(\theta_{t+1}, \mathcal{D}_f)}\right] \geq \mathbb{E}\left[\frac{\lambda_1(\theta_t, \mathcal{D}_f)}{\lambda_d(\theta_t, \mathcal{D}_f)}\right]$, thereby reducing the influence of the forgetting set. The framework introduces two hyperparameters, α and c, which control the relative importance of components (b) and (c) during unlearning. In the following section, we provide a theoretical analysis of how these hyperparameters affect convergence behavior.

4.3. Convergence Guarantee of Influence Eliminating Unlearning Framework

In this section, we provide a convergence guarantee of the proposed unlearning framework on the retaining set.

Theorem 10. *For Φ is a loss function which is L-Lipschitz, $\mu-$strongly and $\beta-$smooth convex, distance between any θ_t generated by Influence Eliminating Unlearning framework is bounded $\frac{||\theta_n - \theta_m||_2}{2} \leq D$, step-size $\eta_t = \frac{1}{\beta}$, the error on retaining set is bounded by:*

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\theta_t, \mathcal{D}_r) - \mathcal{L}(\theta^{\mathcal{D}_r}, \mathcal{D}_r)] \leq\ & LDe^{-\frac{\mu}{\beta}t} \\ & + 2\beta\left(\frac{D}{2}(1-\alpha) + \frac{L}{2\beta}c + \frac{L}{\beta}\right)^2 \\ & + \beta(1-\alpha)^2 + \text{CONST}.\end{aligned} \tag{10}$$

Firstly, we claim that our framework achieves an exponential convergence rate of $\mathcal{O}(e^{-t})$ in t iterations, demonstrating its time efficiency. Secondly, the use of *Gradient*

Ascent and the *Noisy Regularization* component raises the upper error bound with a second-order polynomial of c and α. According to Theorem 4 and Lemma 9, a lower α and an larger c effectively eliminate the forgetting set; however, Theorem 10 demonstrates that this results in a larger upper error bound, which may be harmful to the model's utility.

5. Experiment Setups

In this section, we briefly describe the experimental setups; full details are provided in the Appendix section.

Image Classification. We conduct experiments on CIFAR-10, CIFAR-100 [33], and TINYIMAGENET [36] using ResNet50 [25] and ViT [12] architectures under both random and class-wise data forgetting. We compare three variants of our method (w/`GA`, w/`Noisy`, w/`GA+Noisy`) against four baselines: Fine-tuning (FT), Random Labeling (RL) [21], SCRUB [35], and SALUN [13]. Performance is assessed using accuracy on retaining, forgetting, and testing sets, as well as privacy via MIA. The average performance gap (Avg. Gap) measures the similarity between the performance of the unlearned model and that of a retrained model, with a smaller gap indicating more effective unlearning. We also employ a *relearning convergence delay* metric to quantify how quickly an unlearned model can relearn forgotten data, using gradient descent with varying step-sizes.

Image Generation. We apply unlearning to the latent Stable Diffusion (SD) model [45] to eliminate NSFW content, integrating our `GA` and `Noisy` components with ESD [18] and SALUN [13] baselines. Forgetting is evaluated by generating images from I2P prompts [46] and measuring the ratio of nude images using Nude Detector [2]. Retention is assessed using FID scores by comparing images generated from the IMAGENETTE classes against the corresponding real IMAGENETTE images [29]. To assess vulnerability to relearning, we fine-tune each unlearned model to relearn NSFW concepts and track loss against the original SD v1.4.

6. Experiment Results

In this section, we briefly summarize the experimental results. Complete results and additional ablation studies are provided in the Appendix section.

6.1. Image Classification

Performance Gap. We evaluate our proposed methods against four baseline unlearning approaches under 30% and 50% random and class-wise forgetting scenarios. The results for the ResNet model on the TINYIMAGENET dataset are presented in Tabs. 1 and 2. Additional experimental results on other architectures and datasets are provided in the Appendix section. Across all settings, our methods, especially those using the `Noisy` component, consistently achieve low Avg. Gap scores, indicating strong unlearning performance close to retraining. While `GA` and `Noisy` are effective individually, combining them does not yield further improvement. Compared to baselines, our methods maintain higher accuracy on both retaining and forgetting sets, with slightly worse MIA scores. FT and RL partially reduce influence from the forgetting set but introduce instability or retain residual effects, while SCRUB and SALUN perform poorly overall. Notably, our methods remain robust as the forgetting portion increases, particularly in random data forgetting, where an increasing amount of forgetting presents increased challenges. Overall, these results highlight the effectiveness and robustness of our approach across diverse unlearning settings.

Relearning Risks and Performance Relationship. We analyze the relationship between *relearning convergence delay* $\mathcal{RCD}_{GD}$, which reflects a model's resistance to relearning, and the Avg. Gap, which measures utility. The results for the ResNet architecture on the TINYIMAGENET dataset are shown in Fig. 1. Additional results for other architectures and datasets are included in the Appendix section. Our methods consistently achieve both low Avg. Gaps and high $\mathcal{RCD}_{GD}$ scores across random and class-wise forgetting scenarios, indicating strong performance in preserving utility while limiting the risk of relearning forgotten data. In contrast, baselines such as SALUN achieve high $\mathcal{RCD}_{GD}$ but suffer from poor utility, while FT, RL, and SCRUB show lower Avg. Gaps but are more vulnerable to relearning. These results highlight the ability of our approach to maintain a favorable privacy-utility trade-off.

Ablation Studies about Step-size in Relearning Convergence Delay. We conduct ablation studies to examine the impact of step-size on the *relearning convergence delay* score $\mathcal{RCD}_{GD}$, guided by the theoretical insights from Theorem 4. The results for the ResNet model on the TINYIMAGENET dataset are presented in Fig. 1. Additional experiments on alternative architectures and datasets are provided in the Appendix. Experiments using step-sizes of 10^{-4}, 10^{-5}, and 10^{-6} reveal that while smaller step-sizes slightly increase $\mathcal{RCD}_{GD}$ values, the relative ranking of methods remains consistent. In random forgetting, our methods, particularly those using the `Noisy` component, consistently achieve high $\mathcal{RCD}_{GD}$ scores. On the other hand, FT and SCRUB perform poorly regardless of the step-sizes. In class-wise forgetting, smaller step-sizes compress score ranges, making differentiation harder, though rankings are largely preserved. These results suggest that an appropriately chosen step-size provides a satisfactory balance of sensitivity and computational efficiency when comparing different unlearning methods.

6.2. Image Generation

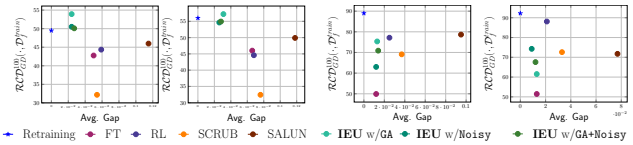
Performance on Forgetting Concepts. We present the Nudity scores for each unlearning method in Tab. 3. The results

Table 1. Performance summary of various unlearning methods for the ResNet model trained on TINYIMAGENET in two unlearning scenarios, 30% random and 50% random data forgetting. Performance gap against Retraining is provided in *(·)*.

Method	Random Data Forgetting (30%)					Random Data Forgetting (50%)				
	$\mathcal{D}_r^{\text{train}}$	$\mathcal{D}_f^{\text{train}}$	$\mathcal{D}^{\text{test}}$	MIA	Avg. Gap	$\mathcal{D}_r^{\text{train}}$	$\mathcal{D}_f^{\text{train}}$	$\mathcal{D}^{\text{test}}$	MIA	Avg. Gap
Retraining	0.930 *(0.000)*	0.470 *(0.000)*	0.546 *(0.000)*	0.874 *(0.000)*	*0.000*	0.950 *(0.000)*	0.429 *(0.000)*	0.495 *(0.000)*	0.831 *(0.000)*	*0.000*
FT	0.888 *(0.042)*	0.609 *(0.139)*	0.562 *(0.015)*	0.870 *(0.003)*	*0.050*	0.897 *(0.052)*	0.595 *(0.165)*	0.538 *(0.043)*	0.830 *(0.001)*	*0.066*
RL	0.760 *(0.170)*	0.512 *(0.042)*	0.522 *(0.024)*	0.875 *(0.001)*	*0.059*	0.786 *(0.163)*	0.525 *(0.096)*	0.507 *(0.013)*	0.833 *(0.002)*	*0.068*
SCRUB	0.909 *(0.021)*	0.642 *(0.172)*	0.570 *(0.024)*	0.874 *(0.000)*	*0.054*	0.910 *(0.039)*	0.632 *(0.203)*	0.556 *(0.061)*	0.831 *(0.000)*	*0.076*
SALUN	0.569 *(0.361)*	0.507 *(0.036)*	0.486 *(0.060)*	0.877 *(0.003)*	*0.115*	0.563 *(0.386)*	0.454 *(0.025)*	0.446 *(0.049)*	0.842 *(0.011)*	*0.118*
IEU w/`GA`	0.916 *(0.013)*	0.547 *(0.077)*	0.541 *(0.005)*	0.873 *(0.001)*	***0.024***	0.924 *(0.026)*	0.525 *(0.095)*	0.494 *(0.001)*	0.830 *(0.002)*	*0.031*
IEU w/`Noisy`	0.881 *(0.049)*	0.508 *(0.038)*	0.540 *(0.006)*	0.869 *(0.004)*	***0.024***	0.900 *(0.050)*	0.483 *(0.054)*	0.494 *(0.000)*	0.829 *(0.002)*	***0.026***
IEU w/`GA+Noisy`	0.885 *(0.045)*	0.518 *(0.048)*	0.536 *(0.010)*	0.869 *(0.005)*	*0.027*	0.898 *(0.052)*	0.486 *(0.056)*	0.495 *(0.000)*	0.828 *(0.003)*	*0.028*

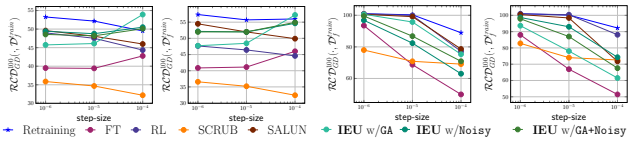
Table 2. Performance summary of various unlearning methods for the ResNet model trained on TINYIMAGENET in two unlearning scenarios, 30% class-wise and 50% class-wise data forgetting. Performance gap against Retraining is provided in *(·)*.

Method	Class-wise Data Forgetting (30%)					Class-wise Data Forgetting (50%)				
	$\mathcal{D}_r^{\text{train}}$	$\mathcal{D}_f^{\text{train}}$	$\mathcal{D}_r^{\text{test}}$	$\mathcal{D}_f^{\text{test}}$	MIA	Avg. Gap	$\mathcal{D}_r^{\text{train}}$	$\mathcal{D}_f^{\text{train}}$	$\mathcal{D}_r^{\text{test}}$	$\mathcal{D}_f^{\text{test}}$	MIA	Avg. Gap
Retraining	0.927 *(0.000)*	0.593 *(0.000)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	*0.000*	0.948 *(0.000)*	0.608 *(0.000)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	*0.000*
FT	0.881 *(0.046)*	0.577 *(0.016)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	***0.012***	0.902 *(0.046)*	0.627 *(0.018)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	*0.013*
RL	0.817 *(0.110)*	0.580 *(0.014)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	*0.025*	0.858 *(0.090)*	0.623 *(0.014)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	*0.021*
SCRUB	0.753 *(0.175)*	0.586 *(0.008)*	0.000 *(0.000)*	0.000 *(0.000)*	0.906 *(0.003)*	*0.037*	0.814 *(0.134)*	0.638 *(0.030)*	0.000 *(0.000)*	0.000 *(0.000)*	0.908 *(0.000)*	*0.033*
SALUN	0.569 *(0.358)*	0.486 *(0.107)*	0.002 *(0.002)*	0.004 *(0.004)*	0.911 *(0.002)*	*0.095*	0.631 *(0.318)*	0.551 *(0.058)*	0.004 *(0.004)*	0.004 *(0.004)*	0.909 *(0.001)*	*0.077*
IEU w/`GA`	0.890 *(0.037)*	0.567 *(0.026)*	0.000 *(0.000)*	0.000 *(0.000)*	0.908 *(0.001)*	*0.013*	0.905 *(0.044)*	0.629 *(0.021)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	*0.013*
IEU w/`Noisy`	0.882 *(0.045)*	0.577 *(0.016)*	0.000 *(0.000)*	0.000 *(0.000)*	0.908 *(0.001)*	***0.012***	0.907 *(0.042)*	0.605 *(0.003)*	0.000 *(0.000)*	0.000 *(0.000)*	0.908 *(0.000)*	***0.009***
IEU w/`GA+Noisy`	0.873 *(0.054)*	0.578 *(0.015)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	*0.014*	0.896 *(0.052)*	0.615 *(0.007)*	0.000 *(0.000)*	0.000 *(0.000)*	0.909 *(0.000)*	*0.012*



(a) Random data forgetting (30%) (b) Random data forgetting (50%) (c) Class-wise data forgetting (30%) (d) Class-wise data forgetting (50%)

Figure 1. Relationship between Avg. Gap and $\mathcal{RCD}_{GD}$ (step-size $\eta = 10^{-4}$) of ResNet model on the training-forgetting dataset $\mathcal{D}_f^{\text{train}}$ of TINYIMAGENET across diverse unlearning scenarios. Our methods consistently achieve a low Avg. Gap and a high $\mathcal{RCD}_{GD}$ score across four forgetting scenarios, demonstrating efficacy in both model utility and privacy.



(a) Random data forgetting (30%) (b) Random data forgetting (50%) (c) Class-wise data forgetting (30%) (d) Class-wise data forgetting (50%)

Figure 2. The $\mathcal{RCD}_{GD}$ values of ResNet model on the training-forgetting set $\mathcal{D}_f^{\text{train}}$ of TINYIMAGENET for various step-sizes. With a small step-size, the $\mathcal{RCD}_{GD}$ values of each unlearning method are less distinguishable, whereas an appropriate step-size enables a significant comparison.

show that the model without using a Noisy component exhibits a high Nudity score, indicating that it is ineffective in eliminating harmful concepts. Meanwhile, incorporating the `Noisy` component shows the effectiveness, while ESD `Noisy` achieves the best safety score in the ESD setting, and SALUN `GA+Noisy` achieves the best safety score in the SALUN setting. These findings suggest that models utilizing the `Noisy` component are significantly less likely to generate harmful content in response to I2P prompts. Additionally, Fig. 3 displays a set of images generated using I2P prompts, highlighting the differences between our proposed approach, the original SD model, and other baseline methods. Please refer to the Appendix for additional generated images from extended ablation studies.

Table 3. Performance of unlearned approaches on IMAGENETTE concepts using FID, I2P prompts using Nudity score, and *relearning convergence delay* of relearning harmful concepts from I2P prompts, measured as $\mathcal{RCD}_{Adam}$.

Method	FID (↓)	Nudity score (↓)	$\mathcal{RCD}_{Adam}$ (↑)
ESD	0.880	0.413	**538.022**
ESD w/`GA`	0.515	0.415	526.653
ESD w/`Noisy`	0.510	**0.224**	524.057
ESD w/`GA+Noisy`	**0.428**	0.330	517.166
SALUN	1.938	0.500	**530.431**
SALUN w/`GA`	0.627	0.337	523.846
SALUN w/`Noisy`	0.742	0.272	525.421
SALUN w/`GA+Noisy`	**0.350**	**0.250**	506.544

Performance on Unrelated Concepts. We present the FID scores for each unlearning method in Tab. 3. The lowest scores achieved by the `GA+Noisy` approaches highlight the effectiveness of our method in preserving generation quality for concepts unrelated to the ones being unlearned. This indicates that our unlearned model produces images that are both more realistic and more aligned with the provided text prompts, as illustrated in Fig. 4. Please refer to the Appendix section for additional generated images from extended ablation studies.

Relearning Convergence Delay. According to the data

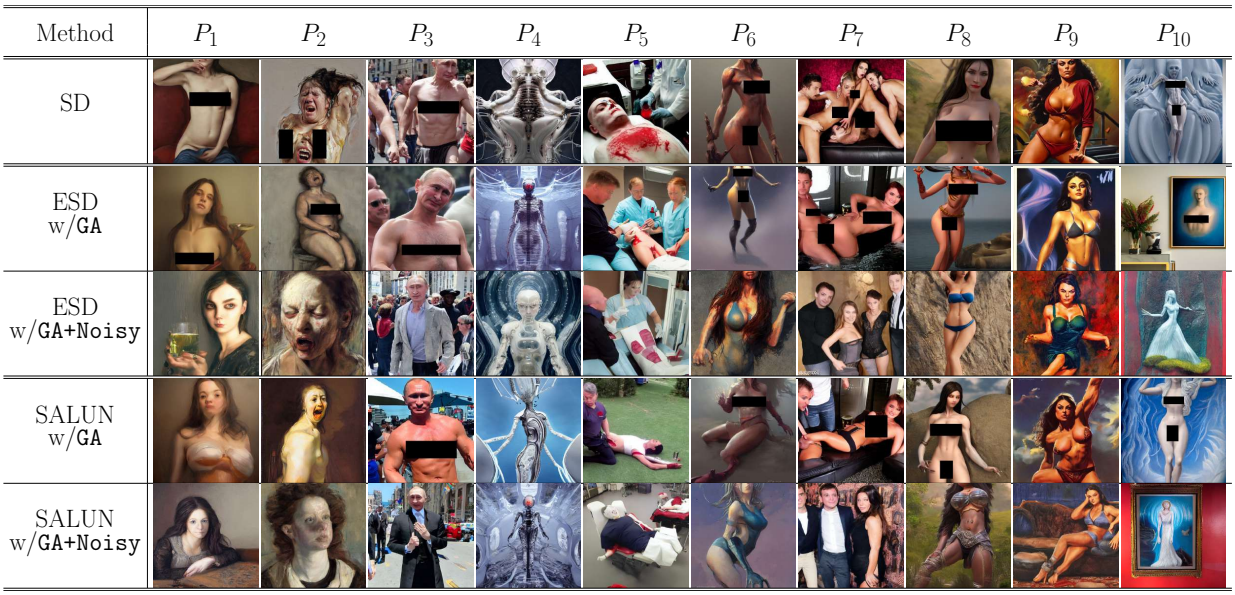


Figure 3. Examples of generated images using various SD models from I2P prompts. The unlearning methods include ESD and SALUN, both with and without the use of Noisy. Each column presents images generated by different SD variants using the same prompt, denoted as P_i. Detailed descriptions of the prompts are provided in the Appendix section.

presented in Tab. 3, we measure the $\mathcal{RCD}$ score to assess how quickly models relearn the forgotten harmful concepts. Although our approach is inspired by the *relearning convergence delay* under gradient descent, SD cannot be directly trained using gradient descent alone; thus, we adopt the Adam optimizer and denote the score as $\mathcal{RCD}_{Adam}$. According to the findings, our methods, particularly those that utilize the Noisy component, exhibit a higher speed of convergence during the relearning process. This suggests that the unlearned model is more susceptible to relearning harmful content. In the meantime, the utilization of both GA+Noisy results in a more rapid convergence, a behavior that contrasts with the GD-motivated setting and appears to be influenced by the dynamics of the Adam optimizer. Following the relearning process, the images that were generated by the model are displayed in the Appendix.

Ablation Studies. We conduct ablation studies to assess the effectiveness of the GA and Noisy components, with results summarized in Tab. 3. The findings show that the combined GA+Noisy outperforms the others. In the ESD setting, GA+Noisy achieves the best FID score and a competitive Nudity score, while in the SALUN setting, it achieves the best FID and Nudity scores. It demonstrates that GA+Noisy is effective in maintaining retention performance and removing harmful concepts. After analyzing the effectiveness of Noisy, we discovered that it has a low Nudity score, indicating less harmful content, as well as a low FID score, indicating more realistic image generation and better consistency in concept retention.

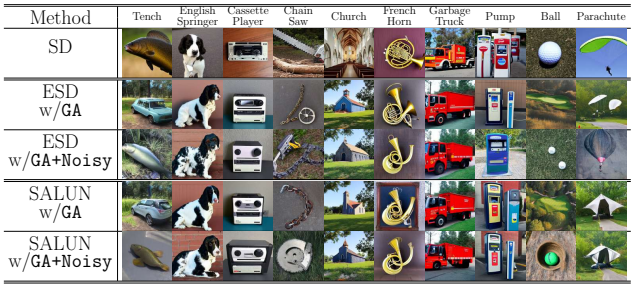


Figure 4. Image generation results for IMAGENETTE classes using models unlearned from I2P harmful concepts. The generated images show that using the Noisy component helps the unlearned model better preserve its performance on unrelated concepts.

7. Conclusion

We presented *relearning convergence delay*, a novel metric that evaluates unlearning effectiveness by measuring how quickly a model relearns forgotten data, capturing both weight-space dynamics and performance on the forgetting set. Based on this insight, we introduced the *Influence Eliminating Unlearning framework*, which combines *Gradient Ascent* and *Noisy Regularization* to eliminate the influence of forgetting data and mitigate the risk of relearning while preserving accuracy on the retaining set. Our experiments and theoretical analysis demonstrate that **IEU**, especially with Noisy component, offers strong unlearning performance, improves resistance to data recovery, and achieves exponential convergence guarantees.

Acknowledgements This work was partly supported by Institute for Information & communication Technology Planning & evaluation (IITP) grants funded by the Korean government MSIT: (RS-2022-II220688, RS-2019-II190421, RS-2024-00437849, RS-2024-00337703). Also, this work was supported by the Cyber Investigation Support Technology Development Program (No.RS-2025-02304983) of the Korea Institute of Police Technology (KIPoT), funded by the Korean National Police Agency. Lastly, this work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00356293).

References

[1] Everton L. Aleixo, J. Colonna, Marco Cristo, and Everlandio Fernandes. Catastrophic forgetting in deep learning: A comprehensive taxonomy. *ArXiv*, abs/2312.10549, 2023. 4, 3

[2] P Bedapudi. Nudenet: Neural nets for nudity classification, detection and selective censoring. *Medium*, 2019. 6, 5

[3] Jordan J. Bird and Ahmad Lotfi. Cifake: Image classification and explainable identification of ai-generated synthetic images. *IEEE Access*, 12:15642–15650, 2023. 1

[4] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. *2015 IEEE Symposium on Security and Privacy*, pages 463–480, 2015. 1

[5] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, A. Terzis, and Florian Tramèr. Membership inference attacks from first principles. *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2021. 2

[6] Oscar Chang, Lampros Flokas, and Hod Lipson. Principled weight initialization for hypernetworks. *ArXiv*, abs/2312.08399, 2020. 5

[7] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7766–7775, 2023. 1, 2, 3

[8] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan S. Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. *ArXiv*, abs/2205.08096, 2022. 3

[9] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan S. Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 18: 2345–2354, 2022. 1

[10] European Commission. The eu artificial intelligence act, 2024. 1

[11] Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *ArXiv*, abs/2410.08827, 2024. 3

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. 6

[13] 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*, abs/2310.12508, 2023. 2, 6, 3, 4

[14] Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. *ArXiv*, abs/2502.05374, 2025. 3

[15] Xiaohua Feng, Jiaming Zhang, Fengyuan Yu, Chengye Wang, Li Zhang, Kaixiang Li, Yuyuan Li, Chaochao Chen, and Jianwei Yin. A survey on generative model unlearning: Fundamentals, taxonomy, evaluation, and future direction. 2025. 1, 2

[16] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Loss-free machine unlearning. *ArXiv*, abs/2402.19308, 2024. 3

[17] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3:128–135, 1999. 4, 3

[18] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2426–2436, 2023. 1, 2, 6, 3, 4

[19] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9301–9309, 2019. 2, 3

[20] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. 3

[21] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *AAAI Conference on Artificial Intelligence*, 2020. 6, 3, 4

[22] SeungBum Ha, Saerom Park, and Sung Whan Yoon. Unlearning's blind spots: Over-unlearning and prototypical relearning attack. *ArXiv*, abs/2506.01318, 2025. 3

[23] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *ArXiv*, abs/1803.01719, 2018. 5

[24] Tomohiro Hayase, Suguru Yasutomi, and Takashi Katoh. Selective forgetting of deep networks at a finer level than samples. *ArXiv*, abs/2012.11849, 2020. 1, 3

[25] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 6

[26] Alvin Heng and Harold Soh. Selective amnesia: A continual learning approach to forgetting in deep generative models. *ArXiv*, abs/2305.10120, 2023. 1

[27] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *ArXiv*, abs/1706.08500, 2017. 2

[28] Seunghoo Hong, Juhun Lee, and Simon S. Woo. All but one: Surgical concept erasing with model preservation in text-to-image diffusion models. In *AAAI Conference on Artificial Intelligence*, 2023. 2, 3

[29] Jeremy Howard and Sylvain Gugger. fastai: A layered api for deep learning. *Inf.*, 11:108, 2020. 6

[30] Abhinav Joshi, Shaswati Saha, Divyaksh Shukla, Sriram Vema, Harsh Jhamtani, Manas Gaur, and Ashutosh Modi. Towards robust evaluation of unlearning in llms via data transformations. In *Conference on Empirical Methods in Natural Language Processing*, 2024. 3

[31] Rohan Kashyap. A survey of deep learning optimizers-first and second order methods. *ArXiv*, abs/2211.15596, 2022. 2

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 2

[33] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. *URl: https://www. cs. toronto. edu/kriz/cifar. html*, 6(1):1, 2009. 6

[34] Siddharth Krishna Kumar. On weight initialization in deep neural networks. *ArXiv*, abs/1704.08863, 2017. 5

[35] Meghdad Kurmanji, P. Triantafillou, and Eleni Triantafillou. Towards unbounded machine unlearning. *ArXiv*, abs/2302.09880, 2023. 2, 6, 3, 4

[36] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. 6

[37] Guihong Li, Hsiang Hsu, Chun-Fu Chen, and Radu Marculescu. Machine unlearning for image-to-image generative models. *ArXiv*, abs/2402.00351, 2024. 1

[38] Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul Grasman, and Eric-Jan Wagenmakers. A tutorial on fisher information. *arXiv: Statistics Theory*, 2017. 3

[39] Aengus Lynch, Phillip Guo, Aidan Ewart, Stephen Casper, and Dylan Hadfield-Menell. Eight methods to evaluate robust unlearning in llms. *ArXiv*, abs/2402.16835, 2024. 3

[40] Thanh Tam Nguyen, Thanh Trung Huynh, Phi-Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *ArXiv*, abs/2209.02299, 2022. 1

[41] OpenAI. Gpt-4 technical report. 2023. 1

[42] Dr. Axel von dem Bussche Paul Voigt. *The EU General Data Protection Regulation (GDPR)*. Springer International Publishing, 1st. edition, 2017. 1

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 4

[44] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Neural Information Processing Systems*, 2019. 4

[45] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 6

[46] Patrick Schramowski, Manuel Brack, Bjorn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22522–22531, 2022. 6

[47] Juwon Seo, Sung-Hoon Lee, Tae-Young Lee, Seungjun Moon, and Gyeong-Moon Park. Generative unlearning for any identity. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9151–9161, 2024. 1

[48] R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2016. 2, 3

[49] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35:13046–13055, 2021. 2, 3

[50] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Neural Information Processing Systems*, 2017. 4

[51] 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*, 46:5362–5383, 2023. 4, 3

[52] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *ArXiv*, abs/2108.11577, 2021. 3

[53] Jing Wu, Trung Le, Munawar Hayat, and Mehrtash Harandi. Erasing undesirable influence in diffusion models. In *Computer Vision and Pattern Recognition*, 2024. 2, 3

[54] Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. Initializing models with larger ones. *ArXiv*, abs/2311.18823, 2023. 5

[55] Thomas Pok-Yin Yu. The essential best and average rate of convergence of the exact line search gradient descent method. 2023. 3

[56] Eric Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1755–1764, 2023. 3

[57] Yushun Zhang, Congliang Chen, Tian Ding, Ziniu Li, Ruoyu Sun, and Zhimin Luo. Why transformers need adam: A hessian perspective. *ArXiv*, abs/2402.16788, 2024. 3, 5