Machine Unlearning in Forgettability Sequence

Junjie Chen[1], **Qian Chen**[1], **Jian Lou**[3],
Xiaoyu Zhang[1], **Kai Wu**[2], **Zilong Wang**[1]*
[1]School of Cyber Engineering, Xidian University, [2]School of Artificial Intelligence, Xidian University,
[3]School of Software Engineering, Sun Yat-Sen University
jjchen0416@stu.xidian.edu.cn, qchen_4@stu.xidian.edu.cn, jian.lou@hoiying.net,
xiaoyuzhang@xidian.edu.cn, kwu@xidian.edu.cn, zlwang@xidian.edu.cn

Abstract

Machine unlearning (MU) is becoming a promising paradigm to achieve the "right to be forgotten", where the training trace of any chosen data points could be eliminated, while maintaining the model utility on general testing samples after unlearning. With the advancement of forgetting research, many fundamental open questions remain unanswered: do different samples exhibit varying levels of difficulty in being forgotten? Further, does the sequence in which samples are forgotten, determined by their respective difficulty levels, influence the performance of forgetting algorithms? In this paper, we identify key factor affecting unlearning difficulty and the performance of unlearning algorithms. We find that samples with higher privacy risks are more likely to be unlearning, indicating that the unlearning difficulty varies among different samples which motives a more precise unlearning mode. Built upon this insight, we propose a general unlearning framework, dubbed RSU, which consists of Ranking module and SeqUnlearn module. RSU is compatible with most of the existing unlearning methods and substantially improves top-performing unlearning algorithms. Extensive experiments are conducted to comprehensively demonstrate RSU's effectiveness in unlearning performance. Overall, we consider our work a significant advancement in deepening the scientific understanding of unlearning and uncovering new avenues for enhancing unlearning algorithms.

Introduction

The significant advancements in data storage and transfer technologies have resulted in an unprecedented increase in the volume of data produced, recorded, and processed. While this abundance of data has facilitated advancements in artificial intelligence (AI), it simultaneously poses risks to user privacy and has contributed to numerous data security breaches. In response to these concerns, regulations such as the General Data Protection Regulation (GDPR) (Voigt and Von dem Bussche 2017) and the California Consumer Privacy Act (CCPA) (Pardau 2018) have been complemented successively to struggle with privacy leakage issues. These regulations aim to empower users by granting them comprehensive control over their data, including the "right to be forgotten," which enables individuals to request the deletion of their personal data collected and stored by various organizations and services. Given its capacity to assess the impact of data on model performance, Machine Unlearning (MU) has investigated to address various trustworthy machine learning challenges. These challenges include defending machine learning security threats (Liu et al. 2022), safeguarding copyright and privacy (Zhang et al. 2024; Eldan and Russinovich 2023), and mitigating data biases to enhance model fairness (Chen et al. 2024; Sattigeri et al. 2022).

As the significance and popularity of MU increase, a diverse range of unlearning algorithms has been developed. While retraining offers the most accurate unlearning strategy, it is also the most computationally intensive. Consequently, developing approximate yet efficient unlearning methods has emerged as a prominent research branch (Graves, Nagisetty, and Ganesh 2021; Warnecke et al. 2021; Thudi et al. 2022a). Although unlearning is garnering increasing attention, it remains a nascent field of research, and the factors influencing the success of various approaches are not yet well understood. Understanding what determines the difficulty of unlearning problems is crucial. In this paper, we focus on investigating the critical factors that influence the particular unlearning performance by posing the following two questions:

- **Q1:** Do different samples exhibit varying levels of difficulty in being forgotten, and how to evaluate the difficulty of forgetting in a more intuitive way?
- **Q2:** Does the sequence in which samples are forgotten, determined by their respective difficulty levels, influence the performance of forgetting algorithms?

Indeed, previous research has demonstrated that the privacy of training data is highly non-uniform. In other words, although data points are generally well protected on average, the empirical risk of privacy leakage resulting from attacks is concentrated on a small fraction of data outliers (Bagdasaryan, Poursaeed, and Shmatikov 2019; Carlini et al. 2022; Feldman and Zhang 2020). Inspired by this discovery, we answer to the **Q1** from the perspective of privacy characteristics of individual data samples. More concretely, we thoroughly delve into the individual data characteristics in unlearning dataset from the perspective of privacy risk of model inversion attack (PRMI), privacy risk of membership inference attack (PRMIA) and difficulty of unlearning

*Corresponding Author

(DU). While both the findings in the literature (Fan et al. 2024) and our observations indicate that the effectiveness of unlearning methods can vary significantly depending on the selection of the unlearning set, we pay more attention to assessing the difficulty of forgetting different data points by exploring the intrinsic connections between privacy risk and unlearning performance. We find that samples more susceptible to model inversion attacks are identical to those with high membership inference attack rate, and also easier to be unlearned. The below figure summaries the high-level characteristics of two types of samples, *i.e.*, the well-chosen samples (the method for picking well-chosen samples is described in details in the later section.) v.s. random-selected samples, with respect to different privacy risks and difficulty of MU.

○ easy/high ▨ hard/low

Samples	PRMI	PRMIA	DU
Well-chosen	○	○	○
Random-selected	▨	▨	▨

Based on our findings, we propose the Ranking and Sequential Unlearning (RSU) algorithm to answer to the **Q2**. Specifically, RSU comprises two modules: the Ranking module and the Sequential Unlearning (SeqUnlearn) module. Firstly, the Ranking module orders the unlearning set from “hard” to “easy” with the increased risk of privacy leakage. We conduct extensive experiments and the results indicate that samples with high privacy risks are more likely to be forgotten and the loss values are positively correlated with privacy risk of membership inference attack. Therefore, we use the loss value as an approximate substitute for privacy risk in the following experiments to measure the difficulty of forgetting. Subsequently, the SeqUnlearn module performs unlearning on the entire unlearning set in a sequential manner. To enhance model performance further, the SeqUnlearn module processes the groups in a hard-to-easy order, mirroring the progressive learning approach used in human curricula. Our comprehensive investigation demonstrates that the RSU framework enhances the unlearning performance of various algorithms and addresses issues identified in our examination of unlearning difficulties. The main contributions can be summarized as follows:

- We analyze the relationship between privacy risk and MU performance, and find that samples more susceptible to model inversion attacks are identical to those with high membership inference attack rate, and are also easier to be forgetton.
- We propose an innovative RSU framework, which is compatible with most of the existing unlearning methods to achieve outstanding performance.
- We conduct extensive experiments to demonstrate that RSU supports efficient unlearning and exhibits superior performance when compatible with state-of-the-art MU algorithms.

Related Work

Machine Unlearning. Machine unlearning techniques (Thudi et al. 2022a; Bourtoule et al. 2021; Guo et al. 2019; Gupta et al. 2021; Thudi et al. 2022b) are designed to eradicate the influence of removed training data (*i.e.*, unlearned data) on the trained model. These techniques are categorized into two types: exact unlearning and approximate unlearning. Exact unlearning involves retraining the model from scratch using the training dataset with the unlearned samples excluded. This approach effectively removes the influence of unlearned data on the model. However, exact unlearning often entails high computational costs, particularly with complex models or large training datasets. To enhance efficiency in machine unlearning, approximate unlearning techniques have been developed. These techniques directly update the trained model’s parameters using information such as gradients calculated from the unlearned data. These methods typically incur significantly lower computational costs than exact unlearning. However, approximate unlearning offers a weaker guarantee of data removal compared to exact unlearning, as the resulting model may still retain some information from the unlearned data.

Model Inversion Attack. Model Inversion (Fredrikson, Jha, and Ristenpart 2015) is an attack that seeks to invert a pretrained model to recover private training data samples. Such attacks enable adversaries to reconstruct high-fidelity data closely resembling the original private training data, raising significant privacy concerns. The significant advancements achieved through generative priors have led to the integration of generative models into the foundational framework for ongoing research on model inversion attacks against deep neural networks (Yuan et al. 2023; Liu et al. 2024b).

Membership Inference Attack. Membership inference attack, one of the most extensively studied privacy attacks, involves an adversary attempting to determine whether a specific example was part of the training dataset. In the context of machine learning (Shokri et al. 2017), the adversary typically has access to a model’s predictions (Nasr, Shokri, and Houmansadr 2018; Sablayrolles et al. 2019), which can range from the complete confidence vector to the label of the class with the highest confidence score (Choquette-Choo et al. 2021). In this study, we employ the Likelihood Ratio Attack (LiRA) (Carlini et al. 2022), detailed in Section Preliminaries, as it demonstrates state-of-the-art performance across all evaluated metrics.

Preliminaries

Consider a prediction problem from some input space $\mathcal{X}$ (e.g., images) to an output space $\mathcal{Y}$ (e.g., labels). We are given training points $z_1, ..., z_n$, where $z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$. Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ be a training dataset of N data points. $\mathcal{D}_u \in \mathcal{D}$ denotes a subset of data that the user needs to unlearn, referred to as the unlearning dataset. Accordingly, the complement of $\mathcal{D}_u$ is the remaining dataset, *i.e.*, $\mathcal{D}_r = \mathcal{D} \backslash \mathcal{D}_u$. For a point z and parameters $\theta \in \Theta$, Let $\mathcal{L}(z, \theta)$ be the loss and let $\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(z_i, \theta)$ be the empirical risk. The empirical risk minimizer is given by $\hat{\theta} =$

$\boldsymbol{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(z_i, \theta)$. We denote θ_0 the original model trained on the entire training set $\mathcal{D}$. Similarly, we denote by θ_u an unlearned model, obtained by a scrubbing algorithm, after removing the influence of $\mathcal{D}_u$ from the trained model θ_0. θ_r is retraining from scratch on an adjusted training dataset that excludes the forget set.

The Likelihood Ration Attack (LiRA). Given a machine learning model $f^* \leftarrow \tau(X_{s^*})$ trained on a dataset $X_{s^*} \in X$, LiRA first trains multiple shadow models $f_s \leftarrow \tau(X_{s^*})$ on random subsets of $X_s \in X$. For a target example $x \in X$, LiRA then computes the logit-gap defined as the difference between highest and second-highest logit $\mathcal{L}(x, f_s^{in})$ for shadow models f_s^{in} that were trained on x, and the logit-gap $\mathcal{L}(x, f_s^{out})$ for shadow models f_s^{out} that were not trained on x. Finally, to predict whether the example x is contained in the training set X_{s^*} of the model f^*, LiRA computes the logit-gap $\mathcal{L}(x, f^*)$ and compares the likelihood of the observed value under the two aforementioned Gaussian distributions. Note that we use LiRA to compute the MIA score in the following experiments.

Computing privacy scores. Given a training dataset X, we can compute a privacy score for each example $x \in X$ by evaluating the average Attack Success Rate (ASR) of our membership inference attacks. Formally, this privacy score is denoted as follows:

$$ASR(x, X) := \underset{f_s \leftarrow \tau(X_s), X_s \leftarrow \mathbb{D}_X}{\mathbf{Pr}} \big[A(x, f_s) = \mathbb{I}[x \in X_s]\big]. \tag{1}$$

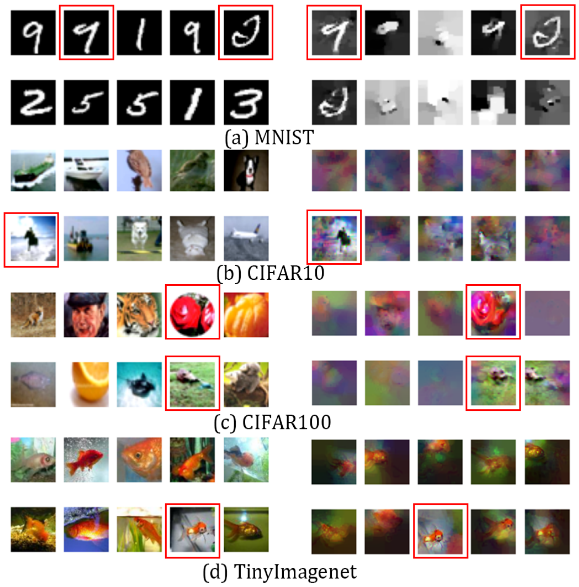
Proxy for unlearning difference. We first introduce a straightforward proxy to compare the unlearning effectiveness of retraining versus other unlearning algorithms. Our objective is to assess the challenge of balancing the removal of data $\mathcal{D}_u$ while maintaining performance on $\mathcal{D}_r$ and ensuring generalization to the test set. We employ a metric to quantify this Tug-of-War (ToW) by measuring the relative difference in accuracy between the unlearned and retrained models across the unlearning, retain, and test sets (Triantafillou et al. 2023).

$$\begin{aligned} ToW(\theta_u, \theta_r, \mathcal{D}_u, \mathcal{D}_r, \mathcal{D}_{val}) &= (1 - da(\theta_u, \theta_r, \mathcal{D}_u)) \\ \cdot (1 - da(\theta_u, \theta_r, \mathcal{D}_r)) &\cdot (1 - da(\theta_u, \theta_r, \mathcal{D}_{val})), \end{aligned} \tag{2}$$

where a is the accuracy on $\mathcal{D}$ of a model f parameterized by θ, $da(\theta_u, \theta_r, \mathcal{D}) = |a(\theta_u, \mathcal{D}) - a(\theta_r, \mathcal{D})|$ is the absolute difference between the accuracy of models θ_u and θ_r on $\mathcal{D}$. Therefore, ToW rewards unlearned models that achieve accuracy levels comparable to those of models retrained from scratch, across the forget, retain, and test sets. ToW values range from 0 to 1, with higher values indicating better unlearning performance.

What Makes Unlearning Effect Different

In this section, we concentrate on comprehensively examining **Q1** and aim to ascertain whether there exist interpretable characteristics of various unlearning examples that significantly influence the difficulty of the unlearning process. First, we investigate and explore this problem from the perspective of data privacy leakage by conducting extensive model inversion attack experiments. As demonstrate in Figure 1, we found a universal phenomenon where under the same experimental settings, some samples can be easily inverted while others are difficult. Then, it raises us a small question: *what are the specific features of samples that are easily inverted, and how do these samples behave when the unlearning request occurs?* To delve deeper into the issue, we select model inversion attack and membership inference attack as an attempt to explore the different behaviors of samples with different privacy sensitivities during unlearning process.


(a) MNIST

(b) CIFAR10

(c) CIFAR100

(d) TinyImagenet

Figure 1: Information leakage from the victims' gradients of 10 images on MNIST, CIFAR-10, CIFAR-100 and TinyImagenet (Red boxes indicate successful attack samples).

Connection between Privacy Risk & MU Performance. We found that samples with higher privacy risks are more likely to be forgotten. Firstly, we selected 100 samples that are easily inverted (well-chosen samples mentioned earlier) and 100 randomly selected samples to explore the connection between privacy risk of model inversion attack and MU performance. We have conducted a large number of experiments on different unlearning algorithms on multiple datasets and multiple networks. The experimental results shown in Table 1 demonstrate that, under the same experimental settings with different data sets and unlearning algorithms, the unlearning accuracy of samples that are easily inverted is lower, e.g., under unlearning method of retrain for CIFAR100, the unlearning accuracy differs by 34% between the two different subsets. We further calculated the membership inference attack (MIA) gap of two different subsets. The MIA gap of samples that are easily inverted is higher similarly, and 0.25 higher than that of random samples.

Additionally, we begin by measuring the average ASR for different samples using the LiRA membership inference attack, and then we selected 100 samples with high ASR

Model (Architecture)	Methods	Original		GA		RL		FT		RfS		Bad-T		L1-sparse	
		MI Low	MI High	MI Low	MI High	MI Low	MI High	MI Low	MI High	MI Low	MI High	MI Low	MI High	MI Low	MI High
MNIST (2 conv. layers 1 FC layer)	$Acc_{D_r}(\%)$	99.99	99.99	99.98	**99.43**	99.86	99.86	99.80	**99.78**	**99.96**	99.97	**99.75**	99.26	99.97	**99.81**
	$Acc_{val}(\%)$	99.26	99.26	99.26	**98.79**	**99.15**	99.23	99.09	**98.99**	99.18	**99.09**	**98.12**	98.39	99.28	**98.34**
	$Acc_{D_u}(\%)$	100	100	100	**89**	100	**92**	100	**94**	100	**88**	100	**88**	100	**87**
	MIA Gap	-	-	0	**0.15**	0.02	**0.27**	0	**0.15**	0	**0.08**	0.23	**0.45**	0	**0.10**
CIFAR10 (Resnet-18)	$Acc_{D_r}(\%)$	100	100	100	**91.91**	99.99	**99.98**	**95.49**	95.62	100	100	99.98	**99.97**	**93.79**	94.12
	$Acc_{val}(\%)$	94.59	94.59	94.5	**85.19**	94.28	**94.24**	**90.30**	90.71	**94.85**	95.35	94.12	**93.65**	89.82	**89.62**
	$Acc_{D_u}(\%)$	100	100	100	**84**	98	**79**	96	**75**	98	**80**	96	**84**	95	**73**
	MIA Gap	-	-	0	**0.19**	0.08	**0.44**	0.22	**0.48**	0.06	**0.26**	0.72	**0.80**	0.27	**0.63**
CIFAR100 (Resnext-50)	$Acc_{D_r}(\%)$	99.98	99.98	98.99	**95.21**	99.97	99.97	93.48	**93.16**	**99.96**	99.98	99.96	99.96	94.15	**92.47**
	$Acc_{val}(\%)$	78.79	78.79	74.53	**71.15**	78.53	**78.46**	71.33	**70.65**	78.02	**78.00**	**77.00**	77.08	71.04	**70.15**
	$Acc_{D_u}(\%)$	100	100	93	**84**	64	**54**	81	**61**	87	**55**	78	**70**	80	**51**
	MIA Gap	-	-	0.08	**0.17**	0.90	0.95	0.41	**0.52**	0.22	**0.47**	0.72	**0.95**	0.33	**0.56**
TinyImagenet (Resnet-50)	$Acc_{D_r}(\%)$	99.98	99.98	99.59	**97.51**	99.98	99.98	93.25	**92.78**	99.98	99.98	99.97	99.97	93.07	**92.72**
	$Acc_{val}(\%)$	59.65	59.65	57.11	**52.06**	**59.38**	59.42	49.72	**49.01**	59.60	**58.22**	**57.62**	57.89	49.45	**49.2**
	$Acc_{D_u}(\%)$	100	100	90	**64**	90	**64**	78	**63**	65	**53**	720	**49**	58	**49**
	MIA Gap	-	-	0.11	**0.45**	0.95	**0.99**	0.51	**0.55**	0.58	**0.69**	0.92	**0.97**	0.56	**0.63**

Table 1: Unlearning performance overview of various MU methods on different samples. MI High refers to samples that are vulnerable to model inversion attacks, while MI Low refers to samples selected randomly. Acc_{D_r}/ Acc_{val}/Acc_{D_u} represent the model's classification accuracy on the the remaining training/validation/ unlearning dataset, respectively. MIA Gap refers to the absolute difference of the MIA score of unlearning from the MIA score of original model. GA, RL, FT, RfS, Bad-T, L1-sparse are baseline unlearning algorithms, and their details elaborated in section of Experiments.

Model (Architecture)	Methods	Original		GA		RL		FT		RfS		Bad-T		L1-sparse	
		MIA Low	MIA High	MIA Low	MIA High	MIA Low	MIA High	MIA Low	MIA High	MIA Low	MIA High	MIA Low	MIA High	MIA Low	MIA High
MNIST (2 conv. layers 1 FC layer)	$Acc_{D_r}(\%)$	99.99	99.99	99.98	**98.98**	**99.88**	99.90	99.91	**99.75**	**99.95**	99.96	99.67	**99.39**	99.85	**99.71**
	$Acc_{val}(\%)$	99.26	99.26	99.25	**98.24**	**99.16**	99.26	99.15	**98.98**	99.19	**99.05**	98.74	**98.35**	98.96	**98.93**
	$Acc_{D_u}(\%)$	100	100	100	**82**	100	**92**	100	**95**	100	**93**	100	**83**	100	**96**
	MIA Gap	-	-	0.02	**0.87**	0.07	**0.93**	0.07	**0.91**	0.05	**0.94**	0.12	**0.82**	0.07	**0.90**
CIFAR10 (Resnet-18)	$Acc_{D_r}(\%)$	100	100	100	**94.86**	99.96	**99.95**	97.66	**96.86**	100	100	100	**99.95**	**94.95**	95.34
	$Acc_{val}(\%)$	94.59	94.59	94.38	**86.86**	**93.79**	93.91	91.14	**90.69**	95.35	**95.25**	94.11	**93.55**	**89.94**	90.05
	$Acc_{D_u}(\%)$	100	100	100	**80**	96	**80**	94	**78**	95	**78**	96	**85**	95	**74**
	MIA Gap	-	-	0.01	**0.28**	0.08	**0.45**	0.25	**0.45**	0.09	**0.31**	0.74	**0.78**	0.23	**0.55**
CIFAR100 (Resnext-50)	$Acc_{D_r}(\%)$	99.98	99.98	99.76	**94.36**	99.98	**99.97**	**88.90**	90.24	99.98	99.98	99.96	99.96	**88.54**	87.81
	$Acc_{val}(\%)$	78.79	78.79	76.92	**69.27**	**77.93**	78.00	**70.61**	71.29	**76.75**	77.84	**76.36**	76.64	70.64	**70.21**
	$Acc_{D_u}(\%)$	100	100	95	**70**	80	**50**	90	**49**	82	**52**	68	**60**	85	**44**
	MIA Gap	-	-	0.04	**0.33**	0.81	**0.96**	0.56	**0.66**	0.27	**0.55**	0.94	**0.97**	0.38	**0.69**
TinyImagenet (Resnet-50)	$Acc_{D_r}(\%)$	99.98	99.98	99.96	**99.51**	99.98	99.98	**89.45**	90.97	99.98	99.98	99.98	**99.97**	**89.17**	90.08
	$Acc_{val}(\%)$	59.65	59.65	58.33	**54.79**	59.58	**59.39**	47.69	**47.95**	58.79	**57.88**	58.37	**57.88**	**48.29** 49.30	
	$Acc_{D_u}(\%)$	100	100	93	**65**	70	**57**	60	**38**	55	**31**	59	**49**	60	**34**
	MIA Gap	-	-	0.03	**0.42**	0.87	**0.98**	0.60	**0.80**	0.66	**0.82**	0.99	**1.00**	0.67	**0.79**

Table 2: Unlearning performance overview of various MU methods on different samples. MIA High refers to samples that are vulnerable to membership inference attack, while MIA Low refers to samples selected randomly.

and 100 randomly selected samples to explore the connection between privacy risk of membership inference attack and MU performance. In the same way, as shown in Table 2, the unlearning accuracy of samples with high ASR drop more and MIA Gap is larger, e.g., the unlearning accuracy differs by 30% and the MIA Gap differs by 0.28 under unlearning method of retrain for CIFAR100. The results shown that samples that are easily inverted or with high ASR are easy-to-forget samples, indicating samples with higher privacy risks are more likely to be forgotten.

Features of easy-to-attack samples. We interpret that easy-to-attack samples are informative, and these are usually atypical, hard, ambiguous, or underrepresented samples. First, we investigate the feature of easy-to-attack samples in the context of model inversion attacks. The adversary in an inversion attack attempts to rebuild or extract training examples by leveraging gradients or model parameters. Based on current knowledge, the generalized features embedded in the gradient or model parameters cannot facilitate the precise reconstruction or extraction of training samples because these features are common. Therefore, the outlier samples may are frequently subject to memorization and, consequently, privacy leakage. Then, we conduct experiments to investigate the relationship between samples that are prone to model inversion attacks and samples that are susceptible to membership inference attacks. Fig.2 (a) reveals that samples that are vulnerable to model inversion attacks exhibit higher ASR scores, indicating that they are more susceptible to membership inference attacks than random samples. Therefore, we collectively referred these samples to easy-to-attack samples. As Fig.2 (b) shown, it is further demonstrated that random samples (MI Low) are usually the least informative, while easy-to-attack (MI High) examples are more informative, by calculating entropy values. A comparison of entropy values between random samples (MIA Low) v.s. easy-to-attack (MIA High) examples can be found in

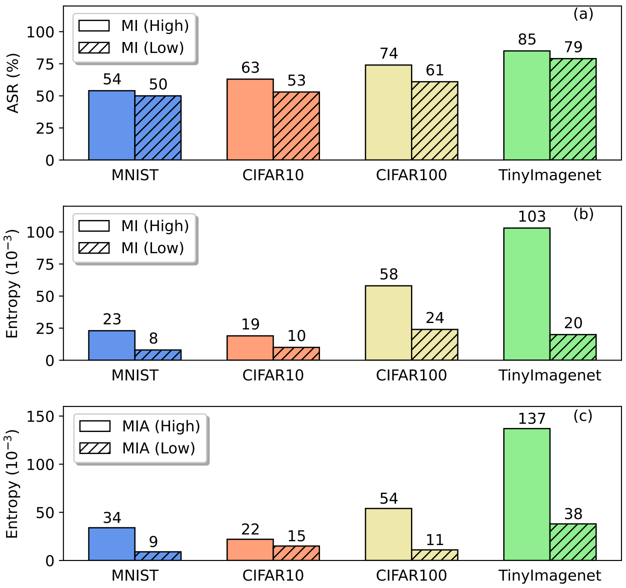


Figure 2: Differences in ASR (a) and in Entropy (b) between easy-to-attack (MI High) and random (MI Low) samples. Differences in Entropy (c) between easy-to-attack (MIA High) and random (MIA Low) samples.

Fig.2 (c).

Class	Top 80%	Bottom 20%	MI (High)	MI (Low)
0	0.9999-0.9984	0.9984-0.6972	0.9981	0.9991
1	0.9999-0.9987	0.9987-0.9702	0.9920	0.9992
2	0.9999-0.9985	0.9985-0.9654	0.9980	0.9992
3	0.9999-0.9985	0.9985-0.9590	0.9983	0.9992
4	0.9999-0.9982	0.9982-0.7755	0.9969	0.9987
5	0.9999-0.9985	0.9985-0.9325	0.9981	0.9989
6	0.9999-0.9984	0.9984-0.9464	0.9974	0.9989
7	0.9999-0.9987	0.9987-0.9293	0.9968	0.9991
8	0.9999-0.9985	0.9985-0.8982	0.9970	0.9990
9	0.9999-0.9987	0.9987-0.8532	0.9984	0.9991

Table 3: The logit distribution of CIFAR10 on different classes (Top 80% refers to the top 80% range of the logit distribution, and Bottom 20% refers to the bottom 20% range of the logit distribution).

In addition, we evaluate the relearn time of MI samples (MI High) v.s. random samples (MI Low) as well as MIA samples (MIA High) v.s. random samples (MIA Low) among several unlearning methods. As Fig.3 and Fig. 4 shown, we find that the relearn time of easy-to-attack samples (MI/MIA High) is greater than that of random samples, indicating that these retained less knowledge of the unlearned data. Besides, we further investigate the data distribution of easy-to-attack samples and random samples. From Table 3, it can be seen that the logit distribution of easy-to-attack samples is in the bottom 20%, commonly known as the tail, while the random samples are distributed in the top 80%.

Insights. Previously, we observed that unlearning algorithms have different behaviours on forget sets with different properties. For example, random unlearning sets are almost trivial to unlearn, the predictions of the model on that example will not change much. On the other hand, for easy-to-attack samples, the predictions between the original and retrained models will differ significantly. These observations suggest that the optimal unlearning algorithm to use is dependent on the properties of the unlearning set. One could therefore pick the different unlearning weights for each unlearning sample, based on these factors. However, in practical scenarios, forget sets may be distributed differently than in our preliminary experiments, that were designed to cleanly separate different factors of interest. Indeed, real-world forget sets would likely contain a mixture of examples from different modes of the data distribution. So, what can be done about these expected heterogeneous forget sets? How can our insights above be leveraged to improve unlearning for such cases? Then we will give a guidebook in next section by proposing Ranking and Sequence Unlearning method (RSU).

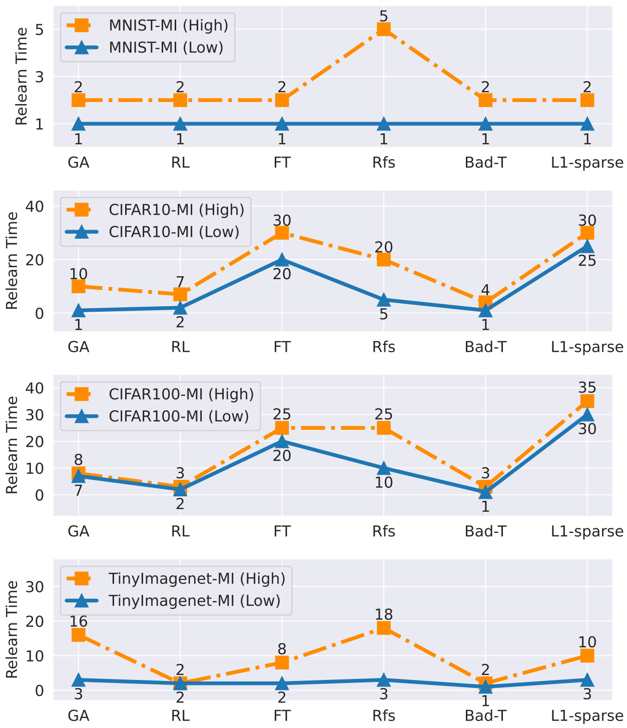


Figure 3: Differences in relearn time between easy-to-attack (MI High) and random (MI Low) samples. A smaller relearn time indicates that the model has retained more knowledge of the unlearned data.

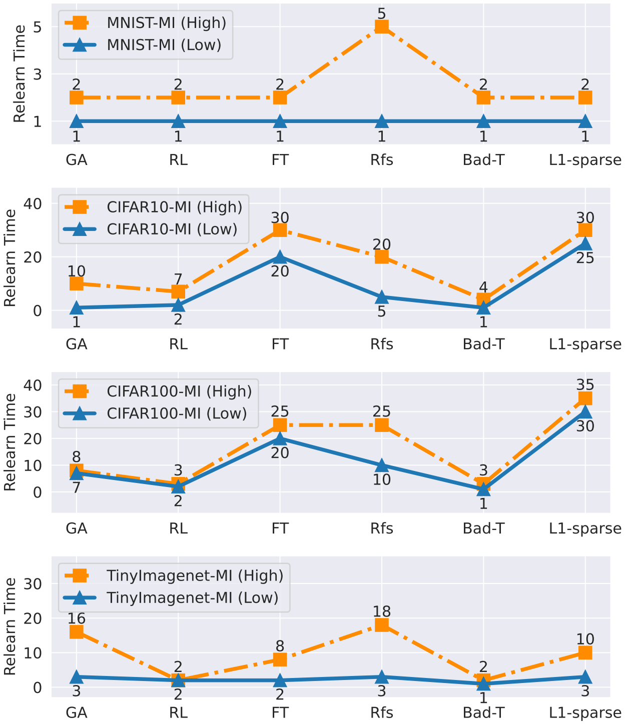


Figure 4: Differences in relearn time between easy-to-attack (MIA High) and random (MIA Low) samples.

Ordering Samples by Forgettability for Improved Unlearning

Overview. Fig. 6 provides a detailed illustration of the RSU framework, which is composed of two key components: the Ranking module and the Sequential Unlearning (SeqUnlearn) module. The framework is designed to optimize the unlearning process by strategically ordering and processing data. As depicted by the black arrows in this figure, the process begins when an unlearning dataset is received. The Ranking module first assesses the difficulty of each data point based on specific criteria, and subsequently ranks the data from hard to easy. This ranking allows the model to prioritize the unlearning of more challenging data points, ensuring that the unlearning process is thorough and effective. Once the data has been ranked, the SeqUnlearn module takes over the following procedure. This module sequentially applies a series of unlearning algorithms to the data, following the order established by the Ranking module. By processing the data in this manner, the SeqUnlearn module mimics a human learning strategy, where more complex tasks are tackled first, gradually moving towards simpler ones. This method enhances the overall efficiency of the unlearning process. Through the coordinated efforts of the Ranking and SeqUnlearn modules, the RSU framework is able to address various challenges in unlearning. The detailed pipeline of RSU can refer to Algorithm 1 .

RSU Framework

Ranking Module. The Ranking module is a critical component of our framework, designed to sort the data from hard to easy based on their loss values. This sorting process is essential for optimizing the unlearning process, as it allows the model to address more challenging data points first. Taking CIFAR10 as an example, we calculated the relationship between the loss values of 10 different intervals and ASR. Results in Fig.5 indicate that the loss values are positively correlated with ASR. Due to the difficulty in calculating sample privacy risks in practical applications, it is cumbersome to use as a metric to measure forgetting difficulty. Therefore, we use the loss values as an approximate alternative. By leveraging loss values, the Ranking module can more effectively prioritize data points during the unlearning process to enhances the efficiency of unlearning algorithms.

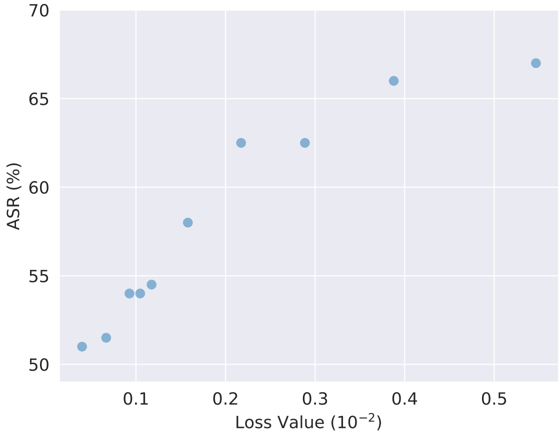


Figure 5: Relationship of loss value and ASR.

SeqUnlearn Module. SeqUnlearn module partition a unlearning dataset $\mathcal{D}_u$ into N subsets such that $\mathcal{D}_i \in \mathcal{D}_{i+1}$ and $\mathcal{D}_n = \mathcal{D}_u$ based on the results of the Ranking module. We regard each subset as a unlearning task. Then we aim to unlearn the model from harder tasks to easier ones. A unlearn task can be considered as hard if the model has a relatively low loss on it, the loss can be obtained at the beginning of unlearning.

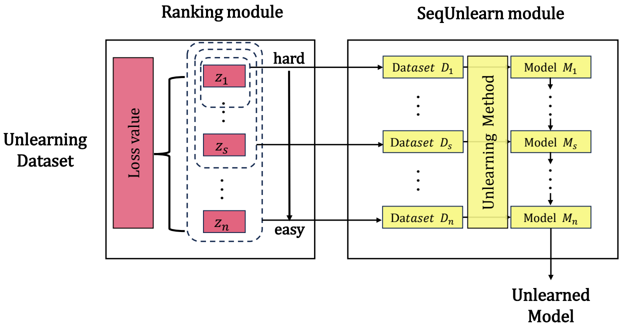


Figure 6: Overview of RSU framework.

SeqUnlearn Module introduces example weight v_i into the above learning objective with an regularizer $\boldsymbol{g}(\boldsymbol{v}, \lambda)$, where $\boldsymbol{v} = [v_1, v_2, ..., v_n]^\intercal \in [0,1]^N$ is a vector of weights, and λ is the age parameter, a hyperparameter which controls the learning pace and determines the proportion of the easiest selected examples at each training epoch. The new learning objective becomes:

$$\min_{\theta; v \in [0,1]^N} \mathbb{E} \sum_{i=1}^{n} v_i l_i + \boldsymbol{g}(\boldsymbol{v}; \lambda). \tag{3}$$

In the SeqUnlearn module, $\boldsymbol{g}(\boldsymbol{v}, \lambda)$ is a negative $l1$-norm:

$$\boldsymbol{g}(\boldsymbol{v}; \lambda) = -\lambda \sum_{i=1}^{N} v_i. \tag{4}$$

In fact, since $\boldsymbol{g}(\boldsymbol{v}, \lambda)$ in Eq. (4) is a convex function of $\boldsymbol{v}$, the global minimum can be easily derived by setting the partial derivative of $\mathbb{E}(\theta, \boldsymbol{v}; \lambda)$ to v_i as zero. Considering $v_i \in [0,1]$, we get the close-formed optimal solution for v^* with the fixed θ,

$$v_i^* = \begin{cases} 1, & l_i < \lambda \\ 0, & \textit{otherwise}. \end{cases} \tag{5}$$

Algorithm 1: **RSU framework**

```
Input: $\mathcal{D}_u = \{x_i, y_i\}_{i=1}^{N}$: unlearning dataset; $f$: the machine unlearning model; $T$: the maximum number of iterations;
Output: $\theta$: the optimal parameters of $f$
Initialize $\theta$, $\boldsymbol{v}$, $\lambda = \lambda_0$, $t = 0$
while $t \neq T$ do
    $t = t + 1$
    Update $\boldsymbol{v}$ by Eq. (5)
    Update $\theta$ by unlearning algorithm
    Update $\lambda$ to a larger value
end while
return $\theta^*$
```

Experiments

Experiment Setup

We evaluate the proposed unlearning method using three public image classification datasets, *i.e.*, MNIST (Deng 2012), CIFAR10, CIFAR100 (Krizhevsky, Hinton et al. 2009), Tiny-ImageNet (Le and Yang 2015). Details of these datasets are provided in Table 4. The selected datasets represent a range of variations in sample size, input dimensionality, and the number of classes. This diversity allows us to examine a range of task complexities: the first two datasets are relatively simple, while the latter two are more complex. Additionally, we utilize three commonly used deep neural networks for image classification, ResNet-18, ResNet-50 (He et al. 2016), and ResNeXt-50 (Xie et al. 2017).

Training details for original models. We empirically evaluate the performance of the proposed unlearning method on four widely used image classification benchmarks. All experiments are conducted using Python 3.8 and PyTorch 1.10.1 on a machine equipped with an Intel Core i9-10980XE CPU, 32 GB of RAM, and an NVIDIA RTX 3090 GPU. The original models are trained for 200 epochs using Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9, weight decay of 5e-4, and an initial learning rate of 0.1, divided by 10 after 100 and 150 epochs, respectively.

Dataset	Dimensionality	Size	# Classes
MNIST	28×28	60000	10
CIFAR-10	$32 \times 32 \times 3$	50000	10
CIFAR-100	$32 \times 32 \times 3$	60000	100
TinyImagenet	$64 \times 64 \times 3$	100000	200

Table 4: Dataset characteristics.

Training details for machine unlearning. To ensure optimal performance, we meticulously tuned the hyperparameters for each unlearning method across various datasets and architectures. For the retrain-from-scratch approach, we adhered to the same training procedure used for the original model, but trained exclusively on the retain set, excluding the forget set. Gradient ascent was trained for 5 epochs, with a learning rate in the range [10^{-4}, 0.01]. Random-label was trained for 10 epochs with a learning rate in the range [0.01, 0.1]. Fine-tune, was trained the model for 10 epochs with a learning rate in the range [0.01, 0.1]. Bad-T was trained for 5 epochs with a learning rate in the range [10^{-4}, 10^{-3}]. L1-sparse was run for 10 epochs with a learning rate in the range [0.01, 0.1], and a sparsity-promoting regularization parameter γ in 10^{-5}.

Evaluation Metric. We utilize some common metrics to evaluate the effectiveness of the unlearning method as follows.

- Acc_{D_r} is the model's classification accuracy on the training set of the remaining dataset.
- Acc_{val} is the model's classification accuracy on the validation dataset.
- Acc_{D_u} is the model's classification accuracy on the training set of the unlearning dataset.
- *Membership inference attack (MIA)* is another metric to assess the efficacy of unlearning. It is achieved by applying the confidence-based MIA predictor to the unlearned model (θ_u) on the forgetting dataset (D_f). The *MIA* success rate can then indicate how many samples in D_u can be correctly predicted as forgetting (i.e., non-training) samples of θ_u. A higher *MIA*-Efficacy implies less information about D_u in θ_u.
- *Relearn time* is an excellent proxy for measuring the amount of unlearned data information left in the model. If a model recovers its performance on unlearned data with just a few steps of retraining, it is extremely probable that the model has retained some knowledge of the unlearned data.
- *ToW* is used to measure the relative difference in accuracy between the unlearned and retrained models across

the unlearning, retain, and test set.

Baseline algorithms. We select the following classic unlearning methods as the baseline algorithms.

- **Retrain from scratch (RfS):** Retrain model from scratch with remaining data points D_r.
- **Gradient Ascent (GA):** GA reverses the model training on D_u by adding the corresponding gradients back to θ_0, i.e., moving θ_0 in the direction of increasing loss for data points to be scrubbed.
- **Random Labels (RL):** Fine-tune the model on D by randomly resampling labels corresponding to images belonging to the unlearning data points D_u.
- **Fine-Tune (FT):** Fine-tune the model on the remaining data D_r using a slightly larger learning rate.
- **Bad-T:** Bad-T introduces a novel machine unlearning method that exploits the student-teacher framework, utilizing both competent and incompetent teachers to induce forgetting in the model (Chundawat et al. 2023).
- **L1-sparse:** L1-sparse incorporates weight sparsity into the unlearning algorithm by applying an L1 penalty during fine-tuning on the retain set, drawing from concepts in the model pruning literature (Liu et al. 2024a).

Model (Architecture)	Methods	Acc_{D_r}	Acc_{val}	Acc_{D_u}	ToW
MNIST (2 conv. layers 1 FC layer)	GA	96.96	86.76	0.00	0.94
	With SPU	**97.92**	**87.52**	0.00	**0.96**
	RL	99.77	87.52	0.00	0.99
	With SPU	**99.88**	**89.25**	0.00	0.99
	BadT	99.05	88.78	0.00	0.98
	With SPU	**99.21**	**88.98**	0.00	**0.99**
CIFAR10 (ResNet-18)	GA	90.98	88.36	0.00	0.85
	With SPU	**94.75**	**82.19**	0.00	**0.90**
	RL	94.09	82.01	0.00	0.89
	With SPU	**96.88**	**83.69**	0.00	**0.94**
	BadT	96.01	81.85	0.00	0.91
	With SPU	**97.57**	**93.19**	0.00	**0.94**
CIFAR100 (ResNext-50)	GA	79.24	53.38	0.00	0.64
	With SPU	**85.49**	**56.85**	0.02	**0.72**
	RL	91.71	65.89	0.00	0.79
	With SPU	**93.74**	**66.15**	0.00	**0.88**
	BadT	76.28	58.90	0.00	0.66
	With SPU	**84.42**	**61.71**	0.04	**0.75**
TinyImagenet (ResNet-50)	GA	50.28	26.16	0.24	0.37
	With SPU	**55.78**	**28.73**	0.22	**0.43**
	RL	90.26	46.55	0.00	0.86
	With SPU	**92.44**	**47.40**	0.00	**0.89**
	BadT	73.58	40.64	0.00	0.66
	With SPU	**75.30**	**42.30**	0.00	**0.68**

Table 5: Performance of various MU methods on original algorithm and with RSU framework considering different unlearning datasets. The unlearning scenario is given by class data forgetting (10% data points across all classes).

Performance

Comparison experiments in class forgetting. Unlearning order plays a pivotal role in enhancing the effectiveness of approximate unlearning methods. We study the impact of unlearning order on the performance of various MU methods in the from hard to easy paradigm. In our experiments, we applied this hard-to-easy paradigm to several MU methods and systematically compared their performance. The performance of the exact unlearning method (Retrain) is also provided for comparison. Note that the better performance of approximate unlearning corresponds to the smaller performance gap with the gold-standard retrained model.

We conduct the comparison experiments for 10% classes of data points removed. The experimental results in these three scenarios (GA, RL and Bad-T) demonstrate the superior effectiveness of RSU. We firstly explore the performance of forgetting class of data samples. we consistently observe that RSU framework improves Acc_{D_r} and Acc_{val} for three different unlearning algorithms and on four different datasets from Table 5. In particular, the performance gap between each approximate unlearning method and Retrain reduces with RSU framework indicating that operating sequentially on classes according to privacy risk can boost the performance of unlearning algorithms. This kind of interclass data distribution makes the difficulty of forgetting between data significantly different compared with random data, thus better reflecting the advantages of the RSU framework. As shown in Table 5 and Table 6, RSU framework achieves a superior performance in class forgetting than random data forgetting.

Comparison experiments in random data forgetting. To thoroughly investigate the impact of the RSU framework on the unlearning process, we selected a subset of 10% of the dataset and apply three unlearning algorithms, *i.e*, GA, RL and BadT. We observe from Table 6 that, for three different unlearning algorithms and on four different datasets, unlearning with RSU outperforms original unlearning, indicating that operating sequentially on subsets according to difficulty of unlearning can boost the performance of unlearning algorithms. However, it is important to note that the effectiveness of the RSU framework becomes more apparent in scenarios involving class data distribution. In cases where the data exhibits significant variation in unlearning difficulty, the RSU framework delivers a substantial performance improvement. This is in contrast to scenarios where the data is randomly distributed. When unlearning is applied to randomly distributed data, the difficulty of forgetting is more uniform across data points, which reduces the potential gains from using the RSU framework. As a result, the performance boost provided by the RSU framework in such cases, while still present, is not as pronounced as in class forgetting. In summary, our experiments validate the effectiveness of the RSU framework in enhancing unlearning algorithms. The framework's sequential approach to handling data, based on unlearning difficulty, consistently improves performance across multiple datasets and algorithms. This makes the RSU framework a versatile and powerful tool for optimizing the unlearning process across a wide range of ap-

Model (Architecture)	Methods	Acc_{D_r}	Acc_{val}	Acc_{D_u}	ToW	*MIA* Gap
MNIST (CNN)	GA	98.7	98.2	98.61	0.97	0.0502
	With SPU	**99.66**	**99.03**	**99.46**	**0.99**	**0.0232**
	RL	98.58	98.47	98.35	0.97	0.2805
	With SPU	**99**	**98.71**	**98.62**	**0.98**	**0.1972**
	BadT	98.33	97.47	98.10	0.96	0.9405
	With SPU	**98.39**	**97.51**	**98.12**	0.96	0.9405
CIFAR10 (ResNet-18)	GA	95.38	88.30	**95.14**	0.89	**0.0198**
	With SPU	**96.83**	**90.14**	96.18	**0.91**	0.0312
	RL	92.69	88.74	89.3	0.82	0.2710
	With SPU	**94.46**	**90.32**	**91.16**	**0.87**	**0.2456**
	BadT	98.61	88.78	89.08	0.87	0.9118
	With SPU	**98.71**	**88.95**	**89.44**	**0.88**	**0.9116**
CIFAR100 (ResNext-50)	GA	76.06	55.49	74.02	0.58	**0.0114**
	With SPU	**77.03**	**56.27**	**75.02**	**0.61**	0.0428
	RL	83.08	69.65	70.62	0.73	0.2240
	With SPU	**87.8**	**70.92**	**74.3**	**0.81**	**0.1736**
	BadT	98.65	69.09	70.28	0.86	0.5598
	With SPU	**98.71**	**69.48**	**70.68**	**0.87**	**0.5596**
TinyImagenet (ResNet-50)	GA	97.08	53.06	94.08	0.56	**0.5580**
	With SPU	**94.67**	**53.60**	**94.28**	**0.57**	0.5909
	RL	92.56	49.46	**61.56**	0.81	**0.0015**
	With SPU	**93.43**	**49.79**	62.01	0.82	0.0164
	BadT	99.48	47.66	54.57	0.89	0.2803
	With SPU	**99.52**	**47.97**	**56.32**	**0.91**	0.2803

Table 6: Performance overview of various MU methods on original algorithm and with RSU framework considering different unlearning datasets. The unlearning scenario is given by random data forgetting (10% data points across all classes), MIA Gap refers to the absolute difference of the MIA score of unlearning from the MIA score of retrain-from-scratch.

plications.

Ablation Study

To thoroughly investigate the impact of forgetting order on unlearning algorithms, we select the GA algorithm within the context of class forgetting and apply this algorithm in three ways: i) Original: *i.e.* applying unlearning algorithm as usual without any specific ordering of the data subsets, ii) Reverse: we deliberately altered the sequence of unlearning by applying the GA algorithm sequentially on data subsets from easy to hard, and iii) RSU: where GA is applied sequentially on the subsets from hard to easy, it prioritizes more challenging data points first. By comparing the results from these three methodologies, we aim to uncover how the order in which data is processed during unlearning influences the success of the unlearning algorithm. We observe from Fig.7 that, for the GA unlearning algorithm and on four different datasets, RSU significantly outperforms Original and Reverse, indicating that operating sequentially on classes according to privacy risk can boost the performance of unlearning algorithms. Interestingly, in some cases, Reverse actually performs worse than Original, indicating that the difficulty of the problem may increase rather than decrease given a poor order strategy. In conclusion, our study demonstrates that the order in which data is forgotten plays a pivotal role in the success of unlearning algorithms.

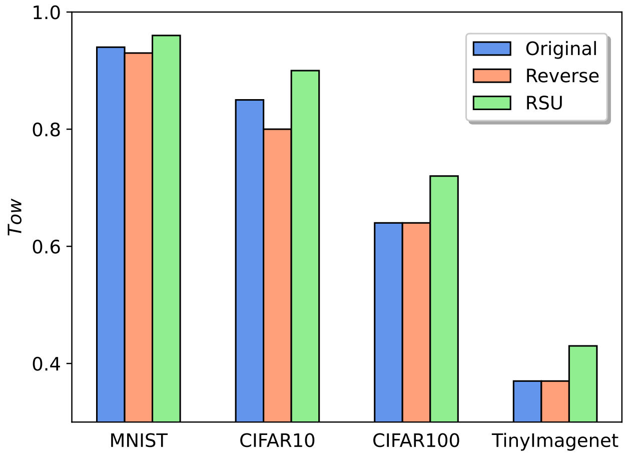


Figure 7: Effect of forgetting order on unlearning algorithm. Original corresponds to unlearning S in one go, whereas Reverse and RSU operate sequentially on subsets of S.

Conclusion

In this paper, we conducted a thorough investigation into the interpretable factors that influence the difficulty of the unlearning process. We found that samples with high privacy risk are more likely to be forgotten, highlighting the relationship between privacy risk and unlearning efficacy. Our investigation led into uncovering previously-unknown behaviours of different unlearning algorithms when considering random forget sets, indicating that different samples exhibit different unlearning difficulty from each other. Based on these insights, we then proposed the RSU framework which consists of Ranking module and SeqUnlearn module. RSU framework can help to prioritize samples that are more challenging to forget and improve the performance of different unlearning algorithms.

References

Bagdasaryan, E.; Poursaeed, O.; and Shmatikov, V. 2019. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems*, 32.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, 141–159. IEEE.

Carlini, N.; Chien, S.; Nasr, M.; Song, S.; Terzis, A.; and Tramer, F. 2022. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, 1897–1914. IEEE.

Chen, R.; Yang, J.; Xiong, H.; Bai, J.; Hu, T.; Hao, J.; Feng, Y.; Zhou, J. T.; Wu, J.; and Liu, Z. 2024. Fast model debias with machine unlearning. *Advances in Neural Information Processing Systems*, 36.

Choquette-Choo, C. A.; Tramer, F.; Carlini, N.; and Papernot, N. 2021. Label-only membership inference attacks. In *International conference on machine learning*, 1964–1974. PMLR.

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

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6): 141–142.

Eldan, R.; and Russinovich, M. 2023. Who's Harry Potter? Approximate Unlearning in LLMs. *arXiv preprint arXiv:2310.02238*.

Fan, C.; Liu, J.; Hero, A.; and Liu, S. 2024. Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. *arXiv preprint arXiv:2403.07362*.

Feldman, V.; and Zhang, C. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33: 2881–2891.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 1322–1333.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11516–11524.

Guo, C.; Goldstein, T.; Hannun, A.; and Van Der Maaten, L. 2019. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*.

Gupta, V.; Jung, C.; Neel, S.; Roth, A.; Sharifi-Malvajerdi, S.; and Waites, C. 2021. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34: 16319–16330.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Liu, J.; Ram, P.; Yao, Y.; Liu, G.; Liu, Y.; SHARMA, P.; Liu, S.; et al. 2024a. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36.

Liu, R.; Wang, D.; Ren, Y.; Wang, Z.; Guo, K.; Qin, Q.; and Liu, X. 2024b. Unstoppable Attack: Label-Only Model Inversion via Conditional Diffusion Model. *IEEE Transactions on Information Forensics and Security*.

Liu, Y.; Fan, M.; Chen, C.; Liu, X.; Ma, Z.; Wang, L.; and Ma, J. 2022. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, 280–289. IEEE.

Nasr, M.; Shokri, R.; and Houmansadr, A. 2018. Comprehensive privacy analysis of deep learning. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, volume 2018, 1–15.

Pardau, S. L. 2018. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23: 68.

Sablayrolles, A.; Douze, M.; Schmid, C.; Ollivier, Y.; and Jégou, H. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, 5558–5567. PMLR.

Sattigeri, P.; Ghosh, S.; Padhi, I.; Dognin, P.; and Varshney, K. R. 2022. Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. *Advances in Neural Information Processing Systems*, 35: 35894–35906.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, 3–18. IEEE.

Thudi, A.; Deza, G.; Chandrasekaran, V.; and Papernot, N. 2022a. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, 303–319. IEEE.

Thudi, A.; Jia, H.; Shumailov, I.; and Papernot, N. 2022b. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, 4007–4022.

Triantafillou, E.; Fabian Pedregosa andJamie Hayes, P. K.; Guyon, I.; Kurmanji, M.; Dziugaite, G. K.; Triantafillou, P.; Zhao, K.; Hosoya, L. S.; Junior, J. C. S. J.; Dumoulin, V.; Mitliagkas, I.; Escalera, S.; Wan, J.; Dane, S.; Demkin, M.; and Reade, W. 2023. NeurIPS 2023 - Machine Unlearning.

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

Warnecke, A.; Pirch, L.; Wressnegger, C.; and Rieck, K. 2021. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Yuan, X.; Chen, K.; Zhang, J.; Zhang, W.; Yu, N.; and Zhang, Y. 2023. Pseudo label-guided model inversion attack via conditional generative adversarial network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 3349–3357.

Zhang, Y.; Zhang, Y.; Yao, Y.; Jia, J.; Liu, J.; Liu, X.; and Liu, S. 2024. Unlearncanvas: A stylized image dataset to benchmark machine unlearning for diffusion models. *arXiv preprint arXiv:2402.11846*.