Roadmap The appendix is composed as follows. Section A presents all the notations and their meaning we use in this paper. Section B presents the rest of the Related Work. Section C gives the proof of our theoretical analysis. Section D gives a more detailed explaination of the proposed algorithm. Section E shows the additional experiment results with more details that are not given in the main paper due to the page limit.

A Notation Table

The notations we use in the paper is summaried in the Table 5.

Table 5: Notation used in this paper

Notations	Description
K	The number of class in the dataset
$\mathcal{D}, \mathcal{X}, \mathcal{Y}$	The general dataset distribution, the feature space and the label space
D	The dataset $D \in \mathcal{D}$
D_{tr}, D_r, D_f	The training set, remaining set and forgetting set
$\Theta_{\mathcal{M}}$	The distribution of models learned using mechanism $\mathcal{M}$
$\boldsymbol{\theta}$	The model weight
$\boldsymbol{\theta}^*$	The optimal model weight
$\boldsymbol{\theta}^*_{f,\text{LS}}$	The optimal model weight trained with D_f whose label is smoothed
$\|\|\boldsymbol{\theta}\|\|$	The 2-norm of the model weight
n	The size of dataset
ε	The up-weighted weight of datapoint z in influence function
$\mathcal{I}(z)$	Influence function of data point z
$h_{\boldsymbol{\theta}}$	A function h parameterized by $\boldsymbol{\theta}$
$\ell(h_{\boldsymbol{\theta}}, z_i)$	Loss of $h_{\boldsymbol{\theta}}(x_i)$ and y_i
$R_{tr}(\boldsymbol{\theta})$	The empirical risk of training set when the model weight is $\boldsymbol{\theta}$
$R_f(\boldsymbol{\theta})$	The empirical risk of forgetting set when the model weight is $\boldsymbol{\theta}$
$R_r(\boldsymbol{\theta})$	The empirical risk of remaining set when the model weight is $\boldsymbol{\theta}$
$H_{\boldsymbol{\theta}}$	The Hessian matrix w.r.t. $\boldsymbol{\theta}$
$\nabla_{\boldsymbol{\theta}}$	The gradient w.r.t. $\boldsymbol{\theta}$
B	Data batch
$B^{\text{LS},\alpha}$	The smoothed batch using α
$z_i = (x_i, y_i)$	A data point z_i whose feature is x_i and label is y_i
$\boldsymbol{y}_i$	The one-hot encoded vector form of y_i
$\boldsymbol{y}_i^{\text{GLS},\alpha}$	The smoothed one-hot encoded vector form of y_i where the smooth rate is α
α	Smooth rate in general label smoothing

B Related Work

Label Smoothing (LS) or positive label smoothing (PLS) (Szegedy et al., 2016) is a commonly used regularization method to improve the model performance. Standard training with one-hot labels will lead to overfitting easily. Empirical studies have shown the effectiveness of LS in noisy label (Szegedy et al., 2016; Pereyra et al., 2017; Vaswani et al., 2017; Chorowski & Jaitly, 2016). In addition, LS shows its capability to reduce overfitting, improve generalization, etc. LS can also improve the model calibration (Müller et al., 2019). However, most of the work about LS is PLS. (Wei et al., 2021) first proposes the concept of negative label smoothing and shows there is a wider feasible domain for the smoothing rate when the rate is negative, expanding the usage of LS.

Influence Function is a classic statistical method to track the impact of one training sample. Koh & Liang (2017) uses a second-order optimization approximation to evaluate the impact of a training sample. Additionally, it can also be used to identify the importance of the training groups (Basu et al., 2020; Koh et al., 2019). The influence function is widely used in many machine-learning tasks. such as data bias solution (Brunet et al., 2019; Kong et al., 2021), fairness (Sattigeri et al., 2022; Wang et al., 2022a), security (Liu et al., 2022a), transfer learning (Jain et al., 2022), out-of-

distribution generalization (Ye et al., 2021), etc. The approach also plays an important role as the algorithm backbone in the MU tasks (Jia et al., 2023; Warnecke et al., 2021; Izzo et al., 2021).

Differential Privacy (DP) is a mathematical framework designed to quantify and mitigate privacy risks in machine learning models. It ensures that the inclusion or exclusion of a single data point in a dataset does not significantly affect the model's output, thus protecting individual data points from being inferred by adversaries Dwork et al. (2006). In machine learning, DP mechanisms such as noise addition and gradient clipping are employed during the training process to provide formal privacy guarantees while maintaining model utility Abadi et al. (2016). These techniques help balance the trade-off between data privacy and model performance, making DP a cornerstone of privacy-preserving machine learning Shokri et al. (2015); McMahan et al. (2018).

A multitude of **privacy risk assessment** tools have been proposed to gauge the degree of leakage associated with the training data. Specifically targeted at the training data, model attacks are often used as a proxy metric for privacy leakage in pretrained models. For example, model inversion attacks are designed to extract aggregate information about specific sub-classes rather than individual samples Fredrikson et al. (2015). Data extraction attacks aim to reverse engineer individual samples used during training Carlini et al. (2020), while property inference attacks focus on inferring properties of the training data Ganju et al. (2018).

More relevant to the current work are **Membership Inference Attacks** (MIA), which predict whether a particular sample was used to train the model. First introduced by Homer et al. Homer et al. (2008), membership attack algorithms were later formalized in the context of DP, enabling privacy attacks and defenses for machine learning models Rahman et al. (2018). Shokri et al. Shokri et al. (2017) introduced MIA based on the assumption of adversarial queries to the target model. By training a reference attack model (shadow model) based on the model inference response, this type of MIA has proven to be powerful in scenarios such as white-box Leino et al. (2019); Nasr et al. (2019); Sablayrolles et al. (2019), black-box Chen et al. (2020); Hisamoto et al. (2019); Song et al. (2020), and label-only Choquette-Choo et al. (2020); Li et al. (2021) access. However, most MIA mechanisms often require training a large number of shadow models with diverse subsets of queries, making them prohibitively expensive. As a result, some recent works have focused on developing cheaper MIA mechanisms Steinke et al. (2023).

Basics of Influence Function Given a dataset $D = \{z_i : (x_i, y_i)\}_{i=1}^n$ and a function h parameterized by $\boldsymbol{\theta}$ which maps from the input feature space $\mathcal{X}$ to the output space $\mathcal{Y}$. Recall the standard empirical risk minimization writes as:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{z \in D} \ell\left(h_{\boldsymbol{\theta}}, z\right). \tag{9}$$

To find the impact of a training point $\hat{z}$, we up-weight its weight by an infinitesimal amount ε[1]. The new model parameter $\boldsymbol{\theta}^{\varepsilon}_{\{\hat{z}\}}$ can be obtained from

$$\boldsymbol{\theta}^{\epsilon_I}_{\{z\}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{z \in D} \ell\left(h_{\boldsymbol{\theta}}, z\right) + \varepsilon \cdot \ell\left(h_{\boldsymbol{\theta}}, \hat{z}\right) \tag{10}$$

When $\varepsilon = -\frac{1}{n}$, it is indicating removing $\hat{z}$. According to Koh & Liang (2017), $\boldsymbol{\theta}^{\varepsilon}_{\{\hat{z}\}}$ can be approximated by using the first-order Taylor series expansion as

$$\boldsymbol{\theta}^{\varepsilon}_{\{\hat{z}\}} \approx \boldsymbol{\theta}^* - \varepsilon \cdot H^{-1}_{\boldsymbol{\theta}^*} \cdot \nabla_{\boldsymbol{\theta}} \ell\left(h_{\boldsymbol{\theta}^*}, \hat{z}\right), \tag{11}$$

where $H_{\boldsymbol{\theta}^*}$ is the Hessian with respect to (w.r.t.) $\boldsymbol{\theta}^*$. The change of $\boldsymbol{\theta}$ due to changing the weight can be given using the influence function $\mathcal{I}(\hat{z})$ as

$$\Delta\boldsymbol{\theta} = \boldsymbol{\theta}^{\varepsilon}_{\{\hat{z}\}} - \boldsymbol{\theta}^* = \mathcal{I}(\hat{z}) = \left.\frac{d\boldsymbol{\theta}^{\varepsilon}_{\{\hat{z}\}}}{d\varepsilon}\right|_{\varepsilon=0} = -H^{-1}_{\boldsymbol{\theta}^*} \cdot \nabla_{\boldsymbol{\theta}} \ell\left(h_{\boldsymbol{\theta}^*}, \hat{z}\right).$$

[1] To distinguish from the ϵ in differential privacy, we use ε here.

C PROOFS

C.1 PROOF FOR THEOREM 1

Proof. For $p(x)$, the Taylor expansion at $x = a$ is

$$p(x) = p(a) + \frac{p'(a)}{1}(x - a) + o \tag{12}$$

Here, $p(\boldsymbol{\theta}) = \nabla R_{tr}(\boldsymbol{\theta}) + \varepsilon \sum_{D_f} \nabla \ell(h_{\boldsymbol{\theta}}, z_i^f)$ so we have

$$p(\boldsymbol{\theta}) = \nabla R_{tr}(a) + \varepsilon \sum_{z^f \in D_f} \nabla \ell(h_a, z^f) + \left[\nabla^2 R_{tr}(a) + \varepsilon \sum_{z^f \in D_f} \nabla^2 \ell(h_a, z^f) \right] (\boldsymbol{\theta} - a) + o \tag{13}$$

For Eq. (2), we expand $p(\boldsymbol{\theta}_f^*)$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_{tr}^*$ as

$$\begin{aligned} p(\boldsymbol{\theta}_f^*) = \nabla R_{tr}(\boldsymbol{\theta}_{tr}^*) + \varepsilon \sum_{z^f \in D_f} \nabla \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^f) \\ + \left[\nabla^2 R_{tr}(\boldsymbol{\theta}_{tr}^*) + \varepsilon \sum_{z^f \in D_f} \nabla^2 \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^f) \right] (\boldsymbol{\theta}_f^* - \boldsymbol{\theta}_{tr}^*) + o = 0 \end{aligned} \tag{14}$$

Since we have $\nabla R_{tr}(\boldsymbol{\theta}_{tr}^*) = 0$ and ignore o, we can get the approximation as

$$\boldsymbol{\theta}_f^* - \boldsymbol{\theta}_{tr}^* \approx - \left[\sum_{z^{tr} \in D_{tr}} \nabla^2 \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^{tr}) + \varepsilon \sum_{z^f \in D_f} \nabla^2 \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^f) \right]^{-1} \left[\varepsilon \sum_{z^f \in D_f} \nabla \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^f) \right] \tag{15}$$

Similarly, we can expand $q(\boldsymbol{\theta}_{tr}^*) = \nabla R_{tr}(\boldsymbol{\theta}_{tr}^*)$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_r^*$ as

$$\begin{aligned} q(\boldsymbol{\theta}_{tr}^*) = \sum_{z^{tr} \in D_{tr}} \nabla \ell(h_{\boldsymbol{\theta}_r^*}, z^{tr}) + \sum_{z^{tr} \in D_{tr}} \nabla^2 \ell(h_{\boldsymbol{\theta}_r^*}, z^{tr})(\boldsymbol{\theta}_{tr}^* - \boldsymbol{\theta}_r^*) \approx 0 \\ \boldsymbol{\theta}_r^* - \boldsymbol{\theta}_{tr}^* \approx \left[\sum_{z^{tr} \in D_{tr}} \nabla^2 \ell(h_{\boldsymbol{\theta}_r^*}, z^{tr}) \right]^{-1} \sum_{z^{tr} \in D_{tr}} \nabla \ell(h_{\boldsymbol{\theta}_r^*}, z^{tr}) \end{aligned} \tag{16}$$

Because of gradient ascent, $\varepsilon = -1$ and we have

$$\begin{aligned} \boldsymbol{\theta}_r^* - \boldsymbol{\theta}_f^* = \boldsymbol{\theta}_r^* - \boldsymbol{\theta}_{tr}^* - (\boldsymbol{\theta}_{tr}^* - \boldsymbol{\theta}_f^*) = \underbrace{\left(\sum_{z^{tr} \in D_{tr}} \nabla^2 \ell(h_{\boldsymbol{\theta}_r^*}, z^{tr}) \right)^{-1} \sum_{z^{tr} \in D_{tr}} \nabla \ell(h_{\boldsymbol{\theta}_r^*}, z^{tr})}_{\Delta \boldsymbol{\theta}_r} \\ \underbrace{- \left(\sum_{z^r \in D_r} \nabla^2 \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^r) \right)^{-1} \sum_{z^f \in D_f} \nabla \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^f)}_{\Delta \boldsymbol{\theta}_f} \end{aligned} \tag{17}$$

Thus, $\|\boldsymbol{\theta}_r^* - \boldsymbol{\theta}_f^*\| = 0$ if and only if $\Delta \boldsymbol{\theta}_f = \Delta \boldsymbol{\theta}_r$, where

$$\sum_{z^{tr} \in D_{tr}} \nabla \ell(h_{\boldsymbol{\theta}_r^*}, z^{tr}) = \underbrace{\left[\sum_{z^{tr} \in D_{tr}} \nabla^2 \ell(h_{\boldsymbol{\theta}_r^*}, z^{tr}) \right] \left[\sum_{z^r \in D_r} \nabla^2 \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^r) \right]^{-1}}_{H(\boldsymbol{\theta}_r^*, \boldsymbol{\theta}_{tr}^*)} \sum_{z^f \in D_f} \nabla \ell(h_{\boldsymbol{\theta}_{tr}^*}, z^f) \tag{18}$$

□

C.2 Proof for Theorem 2

Proof. Recall the loss calculation in label smoothing and we have

$$\ell(h_{\boldsymbol{\theta}}, z^{\text{GLS},\alpha}) = (1 + \frac{1-K}{K}\alpha)\ell(h_{\boldsymbol{\theta}}, (x,y)) + \frac{\alpha}{K}\sum_{y' \in \mathcal{Y}\setminus y} \ell(h_{\boldsymbol{\theta}}, (x,y'))), \tag{19}$$

where we use notations $\ell(h_{\boldsymbol{\theta}}, (x,y)) := \ell(h_{\boldsymbol{\theta}}, z)$ to specify the loss of an example $z = \{x,y\}$ existing in the dataset and $\ell(h_{\boldsymbol{\theta}}, (x,y'))$ to denote the loss of an example when its label is replaced with y'. $\nabla_{\boldsymbol{\theta}}\ell(h_{\boldsymbol{\theta}}, (x,y))$ is the gradient of the target label and $\sum_{y' \in \mathcal{Y}\setminus y} \nabla_{\boldsymbol{\theta}}\ell(h_{\boldsymbol{\theta}}, (x,y'))$ is the sum of the gradient of non-target labels.

With label smoothing in Eq. (19), Eq. (17) becomes

$$\begin{aligned} \boldsymbol{\theta}_r^* - \boldsymbol{\theta}_{f,\text{LS}}^* &\approx \Delta\boldsymbol{\theta}_r + (1 + \frac{1-K}{K}\alpha)\cdot(-\Delta\boldsymbol{\theta}_f) + \frac{1-K}{K}\alpha \cdot \Delta\boldsymbol{\theta}_n \\ &= \Delta\boldsymbol{\theta}_r - \Delta\boldsymbol{\theta}_f + \frac{1-K}{K}\alpha \cdot (\Delta\boldsymbol{\theta}_n - \Delta\boldsymbol{\theta}_f) \end{aligned} \tag{20}$$

where

$$\begin{aligned} \Delta\boldsymbol{\theta}_r &:= \left[\sum_{z^{tr} \in D_{tr}} \nabla^2_{\boldsymbol{\theta}}\ell(h_{\boldsymbol{\theta}_r^*}, z^{tr})\right]^{-1} \sum_{z^{tr} \in D_{tr}} \nabla_{\boldsymbol{\theta}}\ell(h_{\boldsymbol{\theta}_r^*}, z^{tr}) \\ \Delta\boldsymbol{\theta}_f &:= \left[\sum_{z^{r} \in D_{r}} \nabla^2_{\boldsymbol{\theta}}\ell(h_{\boldsymbol{\theta}_{tr}^*}, z^{r})\right]^{-1} \sum_{z^{f} \in D_{f}} \nabla_{\boldsymbol{\theta}}\ell(h_{\boldsymbol{\theta}_{tr}^*}, z^{f}) \end{aligned}$$

as given in Eq. (17). So we have

$$\boldsymbol{\theta}_r^* - \boldsymbol{\theta}_{f,\text{LS}}^* \approx \Delta\boldsymbol{\theta}_r - \Delta\boldsymbol{\theta}_f + \frac{1-K}{K}\alpha \cdot (\Delta\boldsymbol{\theta}_n - \Delta\boldsymbol{\theta}_f) \tag{21}$$

where

$$\Delta\boldsymbol{\theta}_n := \frac{1}{K-1}\left[\sum_{z^{r} \in D_{r}} \nabla^2_{\boldsymbol{\theta}}\ell(h_{\boldsymbol{\theta}_{tr}^*}, z^{r})\right]^{-1} \sum_{z^{f} \in D_{f}} \nabla_{\boldsymbol{\theta}} \sum_{y' \in \mathcal{Y}\setminus y^f} \ell(h_{\boldsymbol{\theta}_{tr}^*}, (x^f, y'))$$

When we have

$$\langle \Delta\boldsymbol{\theta}_r - \Delta\boldsymbol{\theta}_f, \Delta\boldsymbol{\theta}_n - \Delta\boldsymbol{\theta}_f \rangle \leq 0, \tag{22}$$

$\alpha < 0$ can help with MU, making

$$\|\boldsymbol{\theta}_r^* - \boldsymbol{\theta}_{f,\text{NLS}}^*\| \leq \|\boldsymbol{\theta}_r^* - \boldsymbol{\theta}_f^*\| \tag{23}$$

□

C.3 Proof for Theorem 3

Proof. When the optimization is gradient ascent (GA) with negative label smoothing (NLS), Eq. (6) can be written as

$$\ell(h_{\boldsymbol{\theta}}, z^{\text{NLS},\alpha}) = -\left(1 + \frac{1-K}{K}\alpha\right)\cdot\ell(h_{\boldsymbol{\theta}}, (x,y)) - \frac{\alpha}{K}\sum_{y' \in \mathcal{Y}\setminus y} \ell(h_{\boldsymbol{\theta}}, (x,y')), \alpha < 0, \tag{24}$$

Recall $R_{tr}(\boldsymbol{\theta}) = \sum_{z^{tr} \in D_{tr}} \ell(h_{\boldsymbol{\theta}}, z^{tr})$. Denote by $R_f^{\text{NLS}}(\boldsymbol{\theta}; \alpha) = \sum_{z^{\text{LS},\alpha} \in D_f} \ell(h_{\boldsymbol{\theta}}, z^{\text{NLS},\alpha}), \alpha < 0$ the empirical risk of forgetting data with NLS. After MU with label smoothing on D_f by gradient ascent, the resulting model can be seen as minimizing the risk $\gamma_1 \cdot R_{tr}(\boldsymbol{\theta}) - \gamma_2 \cdot R_f^{\text{NLS}}(\boldsymbol{\theta}; \alpha)$, which is a weighted combination of the risk from two phases: 1) machine learning on D_{tr} with weight $\gamma_1 > 0$

and 2) machine unlearning on D_f with weight $\gamma_2 > 0$. Consider an example (x, y) in the forgetting dataset. The loss of this example is:

$$\gamma_1 \ell(h_{\boldsymbol{\theta}}, (x, y)) - \gamma_2 \ell(h_{\boldsymbol{\theta}}, z^{\text{GLS},\alpha}) = \left[\gamma_1 - \gamma_2 \left(1 + \frac{1-K}{K}\alpha\right)\right] \cdot \ell(h_{\boldsymbol{\theta}}, (x, y)) - \frac{\alpha}{K}\gamma_2 \sum_{y' \in \mathcal{Y} \setminus y} \ell(h_{\boldsymbol{\theta}}, (x, y')).$$

When $\left[\gamma_1 - \gamma_2 \left(1 + \frac{1-K}{K}\alpha\right)\right] > 0$, the optimal solution by minimizing this loss is

$$\mathbb{P}(\mathcal{M}(y) = y^{\texttt{pred}}) = \begin{cases} \frac{\gamma_1 - \gamma_2\left(1+\frac{1-K}{K}\alpha\right)}{\left(\gamma_1 - \gamma_2\left(1+\frac{1-K}{K}\alpha\right)\right) - \frac{K-1}{K}\alpha\gamma_2}, & \text{if } y^{\texttt{pred}} = y, \\ \frac{-\frac{\alpha}{K}\cdot\gamma_2}{\left(\gamma_1 - \gamma_2\left(1+\frac{1-K}{K}\alpha\right)\right) - \frac{K-1}{K}\alpha\gamma_2}, & \text{if } y^{\texttt{pred}} \neq y. \end{cases}$$

Accordingly, for another label y', we have

$$\mathbb{P}(\mathcal{M}(y') = y^{\texttt{pred}}) = \begin{cases} \frac{\gamma_1 - \gamma_2\left(1+\frac{1-K}{K}\alpha\right)}{\left(\gamma_1 - \gamma_2\left(1+\frac{1-K}{K}\alpha\right)\right) - \frac{K-1}{K}\alpha\gamma_2}, & \text{if } y^{\texttt{pred}} = y', \\ \frac{-\frac{\alpha}{K}\cdot\gamma_2}{\left(\gamma_1 - \gamma_2\left(1+\frac{1-K}{K}\alpha\right)\right) - \frac{K-1}{K}\alpha\gamma_2}, & \text{if } y^{\texttt{pred}} \neq y'. \end{cases}$$

Then the quotient of two probabilities can be upper bounded by:

$$\log\left(\frac{\mathbb{P}(\mathcal{M}(y) = y^{\texttt{pred}})}{\mathbb{P}(\mathcal{M}(y') = y^{\texttt{pred}})}\right) \leq \left|\log\left(\frac{\gamma_1 - \gamma_2\left(1 + \frac{1-K}{K}\alpha\right)}{-\frac{\alpha}{K}\cdot\gamma_2}\right)\right| = \left|\log\left(\frac{K}{\alpha}(1 - \frac{\gamma_1}{\gamma_2}) + 1 - K\right)\right| = \epsilon.$$

□

D THE DETAILS OF ALGORITHM

D.1 ALGORITHM DETAILS

We provide a more detailed explanation of UGradSL and UGradSL+ in Algorithm 1 here. For UGradSL+, we first sample a batch $B_r = \{z_i^r : (x_i^r, y_i^r)\}_{i=1}^{n_{B_r}}$ from D_r (Line 3-4). Additionally, we sample a batch $B_f = \{z_i^f : (x_i^f, y_i^f)\}_{i=1}^{n_{B_f}}$ from D_f where $n_{B_r} = n_{B_f}$ (Line 5). We compute the distance $d(z_i^r, z_i^f) \in [0, 1]$ for each (z_i^r, z_i^f) pair where $z_i^r \in B_r$ and $z_i^f \in B_f$ (Line 6). For each z_i^f, we count the number of z_i^r whose $d(z_i^r, z_i^f) < \beta$, where β is the distance threshold. This count is denoted by c_i^f (Line 7). Then we get the smooth rate by normalizing the count as $\alpha_i = c_i^f / |B_f|$, where $\alpha_i \in [0, 1]$ (Line 8). GA with NLS is to decrease the model confidence of D_f. The larger the absolute value of α_i, the lower confidence will be given. Our intuition is that a smaller $d(z_i^r, z_i^f)$ means z_i^r is more similar to D_r and the confidence of z_i^f should not be decreased too much.

D.2 ADDITIONAL RESULTS

As mentioned in Section 4, to avoid the smooth rate selection, we propose a self-adaptive smooth rate version. We compare the performance with and without self-adaptive smooth on CIFAR-10 and SVHN. The forgetting scenario is random forgetting. The results are given in Table 8.

E EXPERIMENTS

E.1 ADDITIONAL EXPERIMENTAL SETTINGS

The datasets and model configurations for the original model training and retraining are given in Table 6. We run all the experiments using PyTorch 1.12 on NVIDIA A5000 GPUs and AMD EPYC 7513 32-Core Processor.

The settings of the baseline methods are:

Table 6: The hyperparameters used in the original training and retraining for different models and datasets.

Settings	CIFAR-10 ResNet-18	CIFAR-10 VGG-16	CIFAR-10 ViT	SVHN ResNet-18	CIFAR-100 ResNet-18	ImageNet ResNet-18	20 NewsGroup Bert
Batch Size	256	256	256	256	256	1024	128
Learning rate	$1e^{-2}$	$1e^{-4}$	$1e^{-6}$	$1e^{-2}$	$1e^{-2}$	$1e^{-2}$	$1e^{-4}$
Epochs	160	160	160	160	160	90	60

- Fine-tuning (FT): FT is to fine-tune the original model $\boldsymbol{\theta}_o$ trained from D_{tr} using D_r. We fix the epoch of FT for 10 epochs for all the datasets except ImageNet. We fine-tune ImageNet for 5 epochs. The learning rate is the same as the original training.
- Fisher forgetting (FF): FF is to perturb the $\boldsymbol{\theta}_o$ by adding the Gaussian noise, which with a zero mean and a covariance corresponds to the 4th root of the Fisher Information Matrix with respect to (w.r.t.) $\boldsymbol{\theta}_o$ on D_r (Golatkar et al., 2020). We perform a greedy search for hyperparameter tuning between $1e^{-9}$ and $1e^{-6}$.
- Influence unlearning (IU): IU uses influence function (Koh & Liang, 2017) to estimate the change from $\boldsymbol{\theta}_o$ to $\boldsymbol{\theta}_u$ when one training sample is removed.
- Boundary unlearning[2] (BU): BU unlearns the data by assigning pseudo label and manipulating the decision boundary. It contains boundary shrink and boundary expansion, two types of unlearning methods. The hyper-parameters are the default value in the paper.
- ℓ_1-sparse[3]: ℓ_1-sparse improves machine unlearning by integrating the ℓ_1 norm-based sparse penalty to the loss function. The learning rate is $1e^{-3}$ and we search γ in $[1e^{-5}, 1e^{-1}]$ as given in (Jia et al., 2023).
- SCRUB: SCRUB casts the unlearning problem into a teacher-student framework. We follow the settings exact the same in the original repo[4] where $\gamma = 0.99$ and $\alpha = 0.001$.
- Random Labeling (RL): Unlike FT, RL is to train the model with the random label rather than the fixed label. The settings are the same as for FT.
- SalUN[5]: SalUN takes the weight saliency into consideration. We search γ from $[0.5, 0.9]$.

E.2 Dataset Split of Different Forgetting Paradigms

We also provide the details of dataset split for different forgetting paradigms. For classwise forgetting, we remove the whole class from D_{tr} and D_{te}. In CIFAR-10 and CIFAR-100, the size of D_f is 500 and 5000, respectively. For the other datasets, the size of D_f ranges from the smallest class size to the largest class size because we remove the whole class completely. The selected class to be forgotten is totally random. For random forgetting, we randomly select 10% data from D_{tr} as D_f. We make sure the distribution of D_f is the same as D_{tr}. For CIFAR-20 in group forgetting, each fine-grained class is in the same size which is 500. The coarse class is 2500.

E.3 Additional Class-Wise Forgetting Results

We present the performance of class-wise forgetting in 20 Newsgroup and SVHN datasets in Table 7. The observation is similar in CIFAR-100 and ImageNet given in Table 1. UGradSL and UGradSL+ can improve the MU performance with acceptable time increment, showing the generalization of the proposed method in different modalities and different dataset size.

E.4 Additional Random Forgetting Results

We present the performance of random forgetting in CIFAR-10 and SVHN dataset in Table 8. The observation is similar in CIFAR-100 and Tiny ImageNet given in Table 2.

[2] https://github.com/TY-LEE-KR/Boundary-Unlearning-Code
[3] https://github.com/OPTML-Group/Unlearn-Sparse
[4] https://github.com/meghdadk/SCRUB/tree/main
[5] https://github.com/OPTML-Group/Unlearn-Saliency

Table 7: The experiment results of class-wise forgetting in 20 Newsgroup and SVHN datasets.

20 Newsgroup	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)	RTE (↓, min)
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$98.31_{\pm 2.56}$	$81.95_{\pm 1.69}$	-	26.25
FT	$4.14_{\pm 2.11}$	$9.23_{\pm 3.40}$	$98.83_{\pm 0.86}$	$82.63_{\pm 0.73}$	46.96	1.77
GA	$17.12_{\pm 9.48}$	$62.03_{\pm 5.84}$	$99.99_{\pm 0.01}$	$85.41_{\pm 0.37}$	30.80	**0.37**
IU	$0.00_{\pm 0.00}$	$0.25_{\pm 0.12}$	$100.00_{\pm 0.00}$	$85.58_{\pm 0.20}$	51.27	1.52
BS	$78.33_{\pm 3.47}$	$92.63_{\pm 2.19}$	$97.28_{\pm 0.99}$	$90.93_{\pm 0.81}$	9.76	1.42
UGradSL	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$96.31_{\pm 4.02}$	$78.54_{\pm 5.10}$	1.35	0.39
UGradSL+	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$99.76_{\pm 0.23}$	$84.21_{\pm 0.41}$	**0.93**	2.13
SVHN	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)	RTE (↓, min)
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.01}$	$95.94_{\pm 0.11}$	-	37.05
FT	$6.49_{\pm 1.49}$	$99.98_{\pm 0.04}$	$100.00_{\pm 0.01}$	$96.08_{\pm 0.01}$	23.42	2.42
GA	$87.49_{\pm 1.94}$	$99.85_{\pm 0.09}$	$99.52_{\pm 0.03}$	$95.27_{\pm 0.21}$	3.45	**0.15**
IU	$93.55_{\pm 2.78}$	$100.00_{\pm 0.00}$	$99.54_{\pm 0.03}$	$95.64_{\pm 0.31}$	1.80	0.23
BE	$85.56_{\pm 3.07}$	$99.98_{\pm 0.02}$	$99.55_{\pm 0.01}$	$95.53_{\pm 0.07}$	3.83	3.17
BS	$96.62_{\pm 1.14}$	$99.95_{\pm 0.09}$	$99.99_{\pm 0.00}$	$95.39_{\pm 0.18}$	1.00	3.91
ℓ_1-sparse	$99.78_{\pm 0.31}$	$100.00_{\pm 0.00}$	$98.63_{\pm 0.01}$	$97.36_{\pm 0.18}$	0.75	2.91
RL	$99.99_{\pm 0.01}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$95.44_{\pm 0.13}$	0.13	3.53
EU-k	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$99.61_{\pm 0.08}$	$65.56_{\pm 2.38}$	7.59	4.93
CF-k	$0.09_{\pm 0.03}$	$2.18_{\pm 2.21}$	$99.34_{\pm 0.02}$	$69.87_{\pm 4.13}$	55.88	5.02
SCRUB	$99.99_{\pm 0.02}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$95.79_{\pm 0.26}$	**0.04**	4.97
RL	$99.99_{\pm 0.01}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$95.44_{\pm 0.13}$	0.13	3.53
SalUN	$99.74_{\pm 0.39}$	$100.00_{\pm 0.00}$	$99.53_{\pm 0.02}$	$95.00_{\pm 1.50}$	0.53	4.77
UGradSL	$90.71_{\pm 4.08}$	$99.90_{\pm 0.16}$	$99.54_{\pm 0.04}$	$95.64_{\pm 0.25}$	2.54	0.23
UGradSL+	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$99.82_{\pm 0.62}$	$94.35_{\pm 0.70}$	0.44	4.56

Table 8: The experiment results of random forgetting in CIFAR-10 and SVHN.

CIFAR-10	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)	RTE (↓, min)
Retrain	$8.07_{\pm 0.47}$	$17.41_{\pm 0.69}$	$100.00_{\pm 0.01}$	$91.61_{\pm 0.24}$	-	24.66
FT	$1.10_{\pm 0.19}$	$4.06_{\pm 0.41}$	$99.83_{\pm 0.03}$	$93.70_{\pm 0.10}$	5.65	1.58
GA	$0.56_{\pm 0.01}$	$1.19_{\pm 0.05}$	$99.48_{\pm 0.02}$	$94.55_{\pm 0.05}$	6.80	**0.31**
IU	$17.51_{\pm 2.19}$	$21.39_{\pm 1.70}$	$83.28_{\pm 2.44}$	$78.13_{\pm 2.85}$	10.91	1.18
BE	$0.00_{\pm 0.00}$	$0.26_{\pm 0.02}$	$100.00_{\pm 0.00}$	$95.35_{\pm 0.18}$	7.24	3.17
BS	$0.48_{\pm 0.07}$	$1.16_{\pm 0.04}$	$99.47_{\pm 0.01}$	$94.58_{\pm 0.03}$	6.84	1.41
ℓ_1-sparse	$1.21_{\pm 0.38}$	$4.33_{\pm 0.52}$	$97.39_{\pm 0.31}$	$95.49_{\pm 0.18}$	6.61	1.82
SCRUB	$0.70_{\pm 0.59}$	$3.88_{\pm 1.25}$	$99.59_{\pm 0.34}$	$94.22_{\pm 0.26}$	5.98	4.05
Random Label	$2.80_{\pm 0.37}$	$18.59_{\pm 3.48}$	$99.97_{\pm 0.01}$	$94.08_{\pm 0.12}$	2.24	1.98
UGradSL	$5.87_{\pm 0.51}$	$13.33_{\pm 0.70}$	$98.82_{\pm 0.28}$	$92.17_{\pm 0.23}$	2.01	0.45
UGradSL+	$6.03_{\pm 0.17}$	$10.65_{\pm 0.13}$	$99.79_{\pm 0.03}$	$93.64_{\pm 0.16}$	2.76	3.07
UGradSL (Adp)	$6.04_{\pm 0.11}$	$13.75_{\pm 0.32}$	$99.11_{\pm 0.01}$	$92.07_{\pm 0.02}$	1.76	1.35
UGradSL+ (Adp)	$7.54_{\pm 0.43}$	$13.57_{\pm 0.12}$	$99.67_{\pm 0.00}$	$92.97_{\pm 0.17}$	**1.52**	9.23
SVHN	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)	RTE (↓, min)
Retrain	$4.95_{\pm 0.03}$	$15.59_{\pm 0.93}$	$99.99_{\pm 0.01}$	$95.61_{\pm 0.22}$	-	35.65
FT	$0.45_{\pm 0.14}$	$2.30_{\pm 0.04}$	$99.99_{\pm 0.00}$	$95.78_{\pm 0.01}$	4.49	2.76
GA	$0.58_{\pm 0.04}$	$1.13_{\pm 0.02}$	$99.56_{\pm 0.01}$	$95.62_{\pm 0.01}$	4.86	**0.31**
FF	$0.45_{\pm 0.09}$	$1.30_{\pm 0.12}$	$99.55_{\pm 0.01}$	$95.49_{\pm 0.03}$	4.84	6.02
BE	$0.00_{\pm 0.02}$	$0.02_{\pm 0.17}$	$100.00_{\pm 0.01}$	$96.14_{\pm 0.02}$	5.27	1.03
BS	$0.45_{\pm 0.14}$	$1.13_{\pm 0.05}$	$99.57_{\pm 0.03}$	$95.66_{\pm 0.01}$	4.86	4.24
ℓ_1-sparse	$3.73_{\pm 0.78}$	$8.44_{\pm 0.34}$	$97.84_{\pm 0.28}$	$96.18_{\pm 0.33}$	2.77	0.07
SCRUB	$0.35_{\pm 0.20}$	$4.96_{\pm 0.93}$	$99.94_{\pm 0.02}$	$95.36_{\pm 0.23}$	3.88	3.24
RL	$8.00_{\pm 0.64}$	$29.40_{\pm 11.92}$	$98.72_{\pm 0.45}$	$94.04_{\pm 1.10}$	4.93	1.79
UGradSL	$3.29_{\pm 2.53}$	$14.32_{\pm 4.56}$	$99.89_{\pm 0.02}$	$94.38_{\pm 0.28}$	1.07	0.57
UGradSL+	$5.77_{\pm 2.93}$	$15.95_{\pm 2.26}$	$100.00_{\pm 0.00}$	$95.12_{\pm 0.50}$	0.42	4.44
UGradSL (Adp)	$3.97_{\pm 0.29}$	$14.63_{\pm 2.15}$	$99.89_{\pm 0.01}$	$94.40_{\pm 0.12}$	0.81	2.20
UGradSL+ (Adp)	$5.07_{\pm 0.34}$	$15.89_{\pm 1.03}$	$100.00_{\pm 0.00}$	$95.21_{\pm 0.44}$	**0.21**	14.33

E.5 MU WITH THE OTHER CLASSIFIER

To validate the generalization of the proposed method, we also try the other classification model. We test vision transformer (ViT) and VGG-16 on the task of class-wise forgetting and random forgetting using CIFAR-10, respectively. The results are given in Table 9 and 10. The observation is similar in Table 1 and 2, respectively.

Table 9: The experiment results of class-wise forgetting in CIFAR-10 using ViT.

CIFAR-10	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)	RTE (↓, min)
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$61.41_{\pm 0.81}$	$58.94_{\pm 1.09}$	-	189.08
FT	$3.97_{\pm 0.87}$	$7.60_{\pm 1.76}$	$98.29_{\pm 0.05}$	$80.44_{\pm 0.22}$	61.70	2.99
GA	$33.77_{\pm 6.36}$	$40.47_{\pm 6.63}$	$89.47_{\pm 4.21}$	$71.65_{\pm 2.79}$	41.63	0.32
IU	$1.74_{\pm 0.09}$	$2.16_{\pm 0.61}$	$73.96_{\pm 0.01}$	$68.88_{\pm 0.00}$	54.65	0.24
BE	$85.56_{\pm 3.07}$	$99.98_{\pm 0.02}$	$99.55_{\pm 0.01}$	$95.53_{\pm 0.07}$	22.30	3.17
UGradSL	$68.11_{\pm 11.03}$	$73.84_{\pm 9.58}$	$84.11_{\pm 2.70}$	$68.33_{\pm 1.69}$	22.54	**0.22**
UGradSL+	$99.99_{\pm 0.01}$	$99.99_{\pm 0.02}$	$94.46_{\pm 1.06}$	$77.26_{\pm 1.19}$	**12.85**	5.86

Table 10: The experiment results of random forgetting across all the classes in CIFAR-10 using VGG-16

CIFAR-10	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)	RTE (↓, min)
Retrain	$11.41_{\pm 0.41}$	$11.97_{\pm 0.50}$	$74.65_{\pm 0.23}$	$66.13_{\pm 0.16}$	-	9.48
FT	$1.32_{\pm 0.13}$	$3.48_{\pm 0.13}$	$74.24_{\pm 0.04}$	$67.04_{\pm 0.10}$	4.96	0.60
GA	$1.35_{\pm 0.08}$	$2.18_{\pm 0.66}$	$73.95_{\pm 0.01}$	$66.88_{\pm 0.01}$	5.33	**0.14**
IU	$1.74_{\pm 0.09}$	$2.16_{\pm 0.61}$	$73.96_{\pm 0.01}$	$68.88_{\pm 0.00}$	5.73	0.24
FF	$1.35_{\pm 0.09}$	$2.21_{\pm 0.58}$	$73.95_{\pm 0.02}$	$66.87_{\pm 0.04}$	5.63	1.02
BE	$0.01_{\pm 0.01}$	$0.23_{\pm 0.05}$	$99.98_{\pm 0.00}$	$94.04_{\pm 0.21}$	19.10	1.09
BS	$0.01_{\pm 0.01}$	$0.22_{\pm 0.03}$	$99.98_{\pm 0.01}$	$94.00_{\pm 0.14}$	19.09	3.17
ℓ_1-sparse	$1.27_{\pm 1.13}$	$3.60_{\pm 2.41}$	$98.97_{\pm 1.13}$	$92.18_{\pm 1.46}$	17.22	0.08
SCRUB	$61.16_{\pm 50.89}$	$44.65_{\pm 43.31}$	$39.26_{\pm 50.57}$	$36.95_{\pm 46.68}$	36.75	0.91
UGradSL	$13.45_{\pm 0.63}$	$11.77_{\pm 0.54}$	$65.05_{\pm 0.48}$	$58.52_{\pm 0.38}$	**4.86**	0.19
UGradSL+	$12.41_{\pm 0.32}$	$14.96_{\pm 0.52}$	$65.90_{\pm 0.52}$	$58.58_{\pm 0.35}$	5.13	1.08

E.6 Streisand Effect

From the perspective of security, it is important to make the predicted distributions are almost the same from the forgetting set D_f and the testing set D_{te}, which is called Streisand effect. We investigate this effect in the *random forgetting* on CIFAR-10 by plotting confusion matrix as shown in Figure 3. It can be found that our method will not lead to the extra hint of D_f.

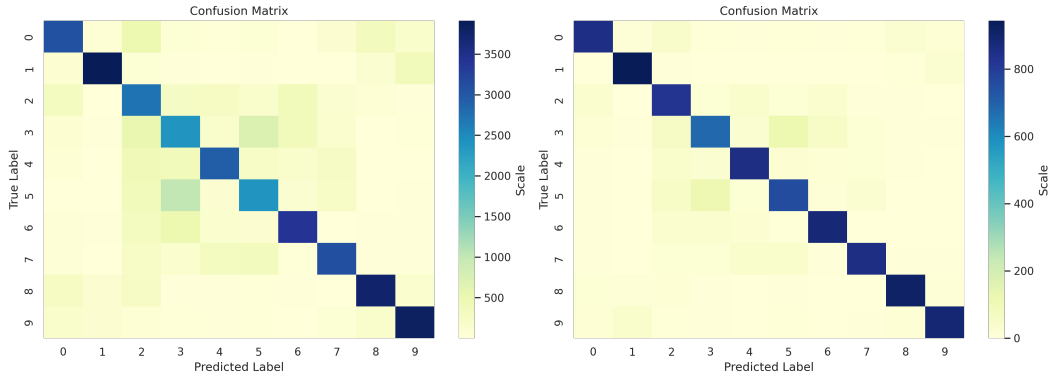

Figure 3: The confusion matrix of testing set and forgetting set D_f using our method on CIFAR-10 with random forgetting across all the classes. There is no big difference between the prediction distribution. Our method will not make D_f more distinguishable.

E.7 Gradient Analysis

As mentioned in Section 3.3, $\langle \Delta\boldsymbol{\theta}_r - \Delta\boldsymbol{\theta}_f, \Delta\boldsymbol{\theta}_n - \Delta\boldsymbol{\theta}_f \rangle \leq 0$ is always practically valid. We practically check the results on CelebA dataset. The distribution of $\langle \Delta\boldsymbol{\theta}_r - \Delta\boldsymbol{\theta}_f, \Delta\boldsymbol{\theta}_n - \Delta\boldsymbol{\theta}_f \rangle$ is shown in Figure 4, which aligns with our assumption.

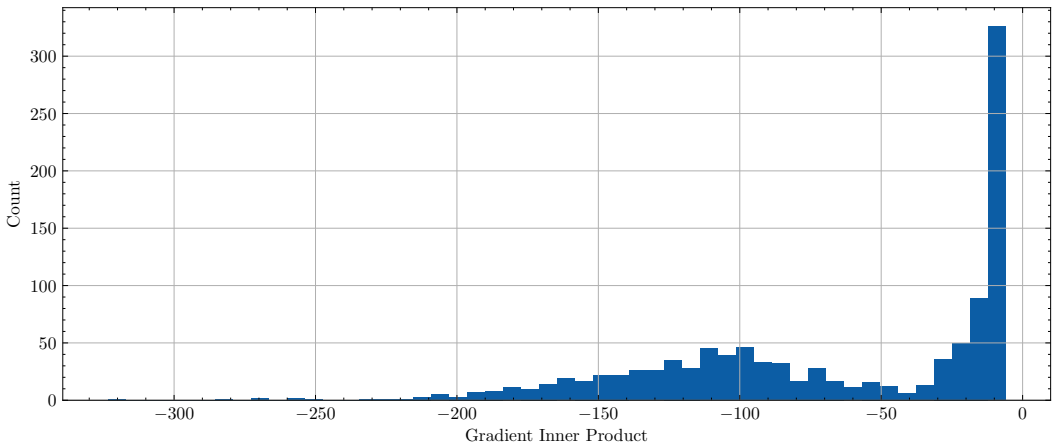

Figure 4: The distribution of $\langle \Delta\boldsymbol{\theta}_r - \Delta\boldsymbol{\theta}_f, \Delta\boldsymbol{\theta}_n - \Delta\boldsymbol{\theta}_f \rangle$ on CelebA dataset.

E.8 The difference between UGradSL and UGradSL+

Although UGradSL and UGradSL+ look similar, the intuition of these two method is totally different because of the difference between FT and GA. We conducted experiments to illustrate the difference between GA and FT as well as UGradSL and UGradSL+. The results are given in Table 11. The dataset and forgetting paradigm is CIFAR-10 random forgetting. It can be found that the difference becomes much larger when the number of epochs is over 8. When the number of epochs is 10, the model is useless because TA is less than 10%. We also report the performance of UGradSL and UGradSL+ in different epochs. For UGradSL, when the epochs are over 14, the model cannot be used at all. For UGradSL+, the algorithm is much more stable, showing the very good adaptive capability.

Table 11: The difference between GA and FT as well as UGradSL and UGradSL+ on CIFAR-10 regarding the number of epochs. The forgetting paradigm is random forgetting.

	Gradient Ascent					Fine-tuning				
Epoch	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)
5	0	0.32	95.31	100	3.98	0.04	0.34	95.13	99.99	3.96
6	0	0.40	95.34	100	3.96	-	-	-	-	-
7	0.82	2.22	93.24	99.26	3.95	-	-	-	-	-
8	3.44	4.78	90.80	96.18	4.03	-	-	-	-	-
9	10.34	12.76	83.42	89.00	7.44	-	-	-	-	-
10	76.26	72.22	6.49	24.24	74.21	0.04	0.24	94.97	99.99	4.02
15	-	-	-	-	-	0.02	0.80	94.68	99.96	3.97
	UGradSL					UGradSL+				
Epoch	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)	UA	$\text{MIA}_{\text{Score}}$	RA	TA	Avg. Gap (↓)
10	14.98	33.22	77.18	84.07	16.51	6.26	14.10	93.39	99.62	1.33
11	24.26	34.38	68.22	75.06	23.61	6.52	11.66	93.04	99.37	1.21
12	28.70	24.62	68.17	74.39	22.46	21.46	27.38	89.41	97.07	10.36
13	38.46	72.90	61.78	64.72	40.99	29.48	31.92	87.74	94.93	14.46
14	99.86	86.74	0.45	0.20	91.26	31.62	32.68	86.53	93.36	15.88
Retrain	4.5	11.62	95.21	100	-	4.5	11.62	95.21	100	-

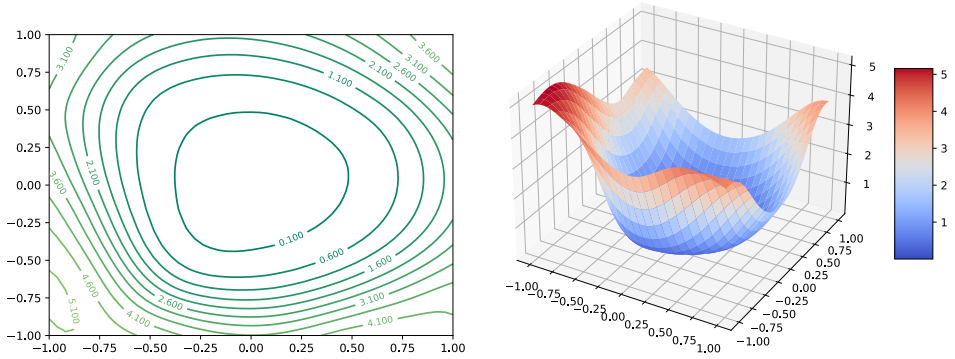

Figure 5: The loss land scape of $\boldsymbol{\theta}_r$ on CIFAR-10 and the model is ResNet-18.

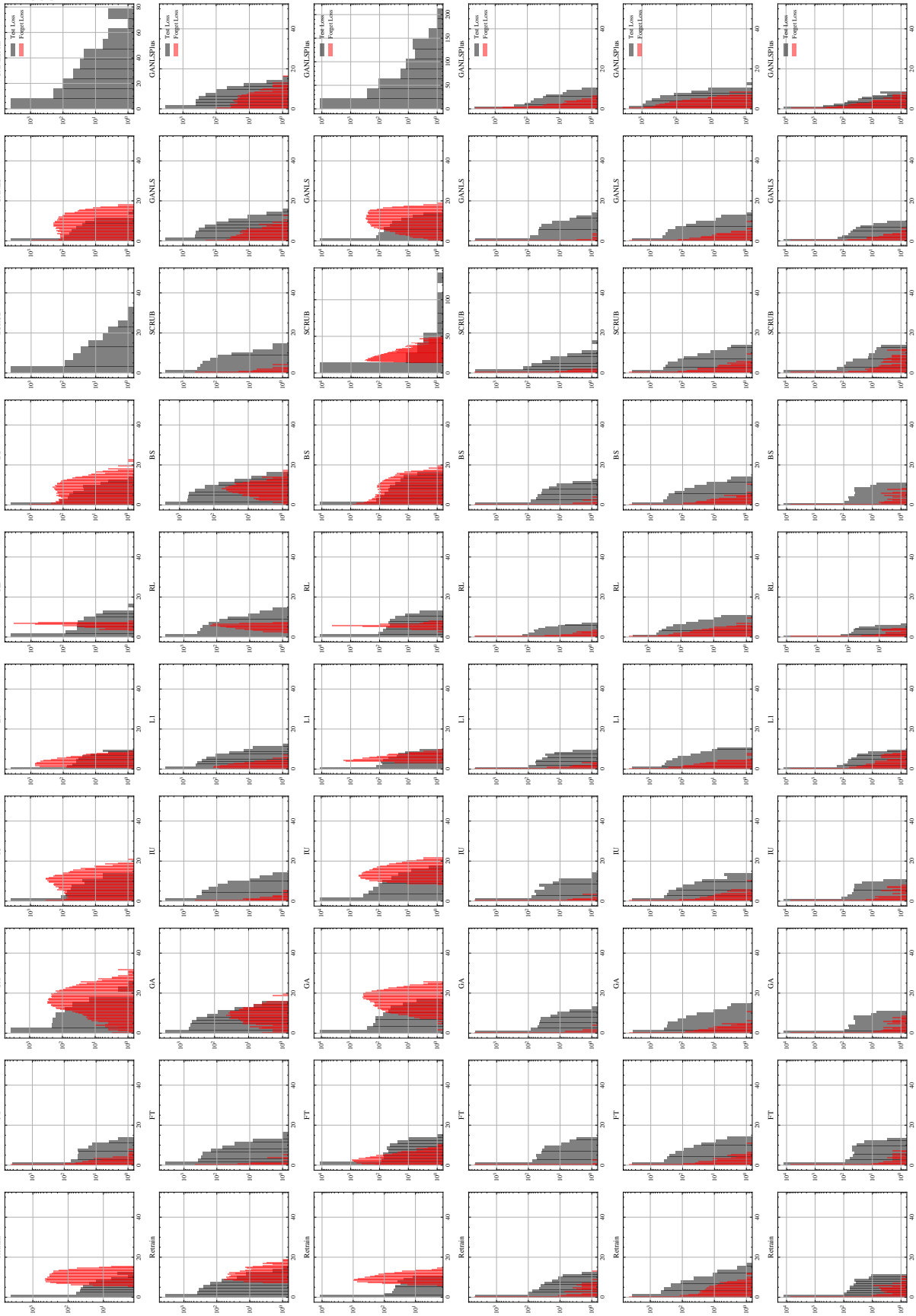

Figure 6: The distributions of the cross-entropy losses for the forget and test instances from the unlearned models. The y-axis is in log scale for better visualization. From the first to the last figure, they are random forgetting on CIFAR-10, CIFAR-100, SVHN and class-wise forgetting on CIFAR-10, CIFAR-100, SVHN.