GDR-GMA: Machine Unlearning via Direction-Rectified and Magnitude-Adjusted Gradients

Anonymous Authors

ABSTRACT

As concerns over privacy protection grow and relevant laws come into effect, machine unlearning (MU) has emerged as a pivotal research area. Due to the complexity of the forgetting data distribution, the sample-wise MU is still open challenges. Gradient ascent, as the inverse of gradient descent, is naturally applied to machine unlearning, which is also the inverse process of machine learning. However, the straightforward gradient ascent MU method suffers from the trade-off between effectiveness, fidelity, and efficiency. In this work, we analyze the gradient ascent MU process from a multi-task learning (MTL) view. This perspective reveals two problems that cause the trade-off, *i.e.,* the gradient direction problem and the gradient dominant problem. To address these problems, we propose a novel MU method, namely GDR-GMA, consisting of Gradient Direction Rectification (GDR) and Gradient Magnitude Adjustment (GMA). For the gradient direction problem, GDR rectifies the direction between the conflicting gradients by projecting a gradient onto the orthonormal plane of the conflicting gradient. For the gradient dominant problem, GMA dynamically adjusts the magnitude of the update gradients by assigning the dynamic magnitude weight parameter to the update gradients. Furthermore, we evaluate GDR-GMA against several baseline methods in three sample-wise MU scenarios: random data forgetting, sub-class forgetting, and class forgetting. Extensive experimental results demonstrate the superior performance of GDR-GMA in effectiveness, fidelity, and efficiency.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**.

KEYWORDS

Deep learning, Machine unlearning, Gradient

1 INTRODUCTION

With the widespread use of artificial intelligence (AI) techniques [2, 12, 13, 16], concerns about privacy protection have escalated. To address these concerns, an increasing number of regulations and laws have been introduced on privacy protection, such as the European Union's GDPR (General Data Protection Regulation) [34]. The GDPR has been promulgated to give people the *right-to-be-forgotten*, which requires information service providers to delete personal data on request from the data owner. Furthermore, this regulation stipulates that providers must also erase the corresponding influence of the requested data. Numerous studies [15, 17, 18, 28, 30, 31] have demonstrated that machine learning models possess the ability to memorize data samples. For example, membership inference attacks [15, 30, 31] can infer whether a data sample is in the training set or not. Consequently, AI service providers must remove the requested data samples and their associated memorized knowledge from the models. However, the naive approach of Retraining the model from Scratch after removing the forgetting data samples from the training set is prohibitively expensive in practice. Therefore, a new research direction for privacy protection emerged to efficiently remove the knowledge of requested data samples, called *machine unlearning (MU)*.

The sample-wise unlearning methods focus on unlearning a subset of data samples in the training set, which can be random data samples, a sub-class of data samples, and a class of data samples. Therefore, it is challenging for the sample-wise MU methods to handle the complex distribution of forgetting data samples. Many works [1, 3, 5, 9–11, 22, 23, 32, 35] attempted to address the sample-wise MU challenges. For example, Bourtoule et al. [1] proposed an exact unlearning method, SISA, to unlearn data samples by retraining the sub-models. However, SISA needs to retrain plenty of sub-models when the requested target data samples are widely distributed across the different shards. To further improve efficiency, many approximate unlearning methods [1, 5, 7, 9–11, 14, 22, 24, 33] were proposed. They estimate the contribution of the forgetting data samples and unlearn them by updating model parameters. Specially, gradient plays an important role in these cutting-edge works. For example, Graves et al. [10] updated the model with the relevant stored gradients. Fan et al. [7] used the ascent gradient to generate the weight salience map.

As one of the basic methods, the straightforward gradient ascent MU method still inspires state-of-the-art works [10, 14, 22, 30, 33], namely the negative gradient (NegGrad). It can be seen as a multi-task learning process: unlearning the forgetting dataset and maintaining the remaining dataset. For image classification, it calculates the ascent gradient of the forgetting data samples and the descent gradient of the remaining data samples to perform a joint model update. Although NegGrad can misclassify the forgetting data samples with few epochs, it will also misclassify most of the remaining samples, leading to a sharp decline in the classification performance (low fidelity). In contrast, the forgetting samples are barely unlearned if NegGrad maintains the classification performance, resulting in MU's low effectiveness and efficiency. Therefore, it suffers from the trade-off between effectiveness, fidelity, and efficiency.

In this work, we identify two key factors for this trade-off problem, *i.e.,* **gradient direction problem** and **gradient domination problem**, as shown in Sec. 3.3 and Sec. 3.4. For the gradient direction problem, we identify two pairs of direction conflicts: 1) the

Unpublished working draft. Not for distribution.

ACM MM, 2024, Melbourne, Australia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ascent gradient of the forgetting data samples and the descent gradient of the remaining data samples; 2) the descent gradient of the remaining data samples and the ascent gradient of the other forgetting data samples (from the difference set of the current forgetting samples and the forgetting set). For the gradient domination problem, we find that a single gradient always dominates the update process, so the MU process solely tends to one task.

To address the above problems and to better apply the gradient ascent to the sample-wise machine unlearning task, we propose a novel unlearning method consisting of Gradient Direction Rectification (GDR) and Gradient Magnitude Adjustment (GMA), dubbed GDR-GMA. For the gradient direction problem, GDR rectifies the direction of gradients to be orthogonal to the conflict gradients. For the gradient domination problem, GMA dynamically adjusts the magnitude of update gradients to balance the two tasks in model updating. We summarize our contributions to this paper as follows:

- We analyze the problems in the straightforward gradient ascent MU method that causes the trade-off among effectiveness, fidelity, and efficiency from a multi-task learning view. This perspective reveals two problems: the gradient direction problem and the gradient domination problem.
- We propose the GDR-GMA unlearning method. For the gradient direction problem to address these two problems, GDR-GMA rectifies the direction between the conflicting gradients by projecting a gradient onto the orthonormal plane of the conflicting gradient. For the gradient domination problem, GDR-GMA dynamically adjusts the magnitude of the update gradients by assigning the dynamic magnitude weight parameter to the update gradients.
- We conduct extensive experiments in three sample-wise MU scenarios: random data forgetting, sub-class forgetting, and class forgetting. Compared to 11 baseline MU methods, GDR-GMA achieves a superior performance in effectiveness, efficiency, and fidelity. Furthermore, we also show the scalability to the existing methods and other computer vision tasks.

2 RELATED WORK

Multi-task learning. Multi-task learning (MTL) aims to help improve the model performance by leveraging the commonalities and differences across multiple tasks. Instead of training on a single task, the model is trained simultaneously on multiple related tasks. In MTL, reducing the direction conflict and domination by a single task is an important topic. For example, Chen et al. [4] normalized the gradient to adaptive balance the loss among tasks. Yu et al. [36] pointed out that the gradient direction conflict and single task domination may damage the model performance. Liu et al. [25] proposed a Multi-Task Attention Network to improve performance. Although MU is not a traditional MTL task, we can utilize the philosophy behind MTL to mitigate the direction and gradient dominant problems of the update gradients in MU.

Class-wise machine unlearning. As a particular case of the sample-wise MU, the class-wise machine unlearning only focuses on forgetting the entire class of data samples. For example, Lin et al. [23] added an entanglement-reduced structure into the model and then transferred the knowledge of the remaining data classes to the unlearned model. Chen et al. [3] shifted the decision boundary of the trained model to forget specific classes.

Sample-wise machine unlearning. The existing sample-wise MU works can be divided into two groups, *i.e., exact machine unlearning* and *approximate unlearning*. Bourtoule et al. [1] proposed SISA to unlearn a subset of the forgetting data samples by retraining the submodels. Even though SISA improves efficiency compared to Retrain, plenty of sub-models still need to be retrained if the forgetting data samples are scattered. To improve the efficiency of exact MU methods, many state-of-the-art works [5, 9, 10, 22, 26, 27, 35] focus on approximate machine unlearning methods. For example, Golatkar et al. [9] and Mehta et al. [27] used the Fisher Information [26] to estimate the contributions. Foster et al. [8] used the Fisher information matrix to select and dampen the important parameter for the forgetting set. Moreover, Chundawat et al. [5] constructed a teacher-student model for MU. Graves et al. [10] stored the gradients related to the target unlearning data during the training process and then subtracted the gradients to update the model's parameters. Liu et al. [24] demonstrated that model sparsity can improve unlearning performance. Fan et al. [7] used the ascent gradients to construct a weight salience map to update only the specific weights rather than the entire model.

Different from the existing methods, we further analyze problems in the straightforward gradient ascent MU method. We identify two key reasons: gradient direction and gradient domination. Furthermore, we propose Gradient Direction Rectification (GDR) and Gradient Magnitude Adjustment (GMA) to address these problems, leading to an effective, fidelity, and efficient MU method.

3 PRELIMINARIES & PROBLEM ANALYSIS

In this section, we will first give a formulation of the sample-wise machine unlearning in Sec. 3.1 and describe the negative gradient method in Sec. 3.2. Then, we will analyze the gradient direction and gradient domination problems in Sec. 3.3 and Sec. 3.4, respectively.

3.1 Sample-wise Machine Unlearning

We first assume a sample space $\mathcal{X} \subseteq \mathbb{R}^d$, the corresponding ground truth labels $\mathcal{Y} = \{1, 2, \ldots, C\}$ (C is the number of data classes), and a training set $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{y})\}$, in which $\boldsymbol{x} \subseteq \mathcal{X}$ and $\boldsymbol{y} \subseteq \mathcal{Y}$. We further define a forgetting dataset $\mathcal{D}_f \subseteq \mathcal{D}$, a remaining dataset $\mathcal{D}_r = \mathcal{D} \backslash \mathcal{D}_f$, and an original model with trainable parameters $\boldsymbol{\omega}_0$, which trained on $\mathcal{D}$.

Definition 1. We define a machine learning algorithm, $\mathcal{A} : \mathcal{D} \to \boldsymbol{\omega}_0$ and a machine unlearning method, $\mathcal{U} : \boldsymbol{\omega}_0 \times \mathcal{D}_r \times \mathcal{D}_f \to \boldsymbol{\omega}_u$. We denote the model that performed the unlearning operation as $\boldsymbol{\omega}_u$ and the model trained with the remaining set $\mathcal{D}_r$ as $\boldsymbol{\omega}_r$. The goal of MU is to attain an unlearned model $\boldsymbol{\omega}_u$, wherein the knowledge of the forgetting data samples $\mathcal{D}_f$ equals to that of the retrained model $\boldsymbol{\omega}_r$. We can formulate it as:

$$\mathcal{K}(\mathcal{D}_f; \boldsymbol{\omega}_u) = \mathcal{K}(\mathcal{D}_f; \boldsymbol{\omega}_r), \tag{1}$$

in which $\mathcal{K}(\cdot)$ is the knowledge measuring function. Note that in the sample-wise machine unlearning task, the forgetting dataset can be a random subset of the training set, a subclass of data samples in a super-class, or a class of data samples.

3.2 Straightforward Gradient Ascent

Here, we will introduce the straightforward gradient ascent MU method, namely NegGrad. First, we perform forward propagation with a forgetting data sample $(\boldsymbol{x}_f, \boldsymbol{y}_f) \subseteq \mathcal{D}_f$, resulting in a prediction vector $\tilde{\boldsymbol{y}}$. We calculate the cross entropy (CE) loss between the prediction vector $\tilde{\boldsymbol{y}}_f$ and the ground truth label $\boldsymbol{y}_f$. Then, we perform a backward propagation to calculate the gradient $\boldsymbol{g}_f = \nabla_\omega \mathcal{L}_{\text{CE}}(\omega; \boldsymbol{x}_f, \boldsymbol{y}_f)$. Again, we can obtain the gradient $\boldsymbol{g}_r$ of a remaining data sample $(\boldsymbol{x}_r, \boldsymbol{y}_r) \subseteq \mathcal{D}_r$. Finally, NegGrad updates the original model parameters as follows:

$$\omega_u \leftarrow \omega_0 + \eta(\boldsymbol{g}_f - \boldsymbol{g}_r), \tag{2}$$

in which η is the learning rate. This process can be seen as multi-task learning. NegGrad has two main tasks: unlearning the forgetting dataset $\mathcal{D}_f$ and maintaining the remaining dataset $\mathcal{D}_r$. Specifically, the ascending gradient $\boldsymbol{g}_f$ is for the unlearning task, and the descent gradient $\boldsymbol{g}_r$ is for the maintaining task.

3.3 Gradient Direction Problem

Here, we will analyze the gradient direction problem. First, we introduce an additional ascent gradient $\boldsymbol{g}_F = \nabla_\omega \mathcal{L}_{\text{CE}}(\omega; \boldsymbol{x}_F, \boldsymbol{y}_F)$ of the other forgetting data samples $\boldsymbol{x}_F \subseteq \mathcal{D}_f \backslash \boldsymbol{x}_f$. Then, we define the gradient direction conflict as follows:

Definition 2. Given φ_{ij} is the angle of two gradients $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$. The two gradients have direction conflicts if $\cos \varphi_{ij} < 0$.

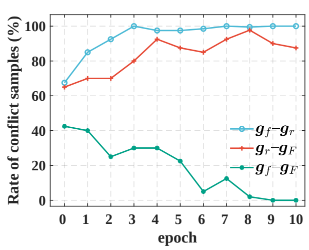


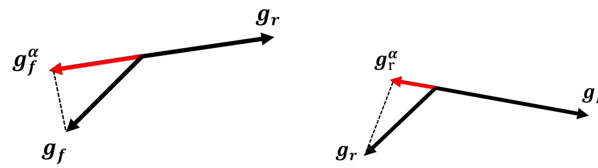
Figure 1: Measurement of the percentage conflicting samples on CIFAR-10 using ResNet-18. The percentage maintains high for $\boldsymbol{g}_f$−$\boldsymbol{g}_r$ (blue) and $\boldsymbol{g}_r$−$\boldsymbol{g}_F$ (red). The percentage tends to be low for $\boldsymbol{g}_f$−$\boldsymbol{g}_F$ (green).

During the NegGrad unlearning process, we have three pairs of gradients as follows:

- $\boldsymbol{g}_f$−$\boldsymbol{g}_r$: the ascent gradient $\boldsymbol{g}_f$ of the forgetting samples and the descent gradient $\boldsymbol{g}_r$ of the remaining samples;
- $\boldsymbol{g}_r$−$\boldsymbol{g}_F$: the descent gradient $\boldsymbol{g}_r$ of the remaining samples and the ascent gradient $\boldsymbol{g}_F$ of the other forgetting samples.
- $\boldsymbol{g}_f$−$\boldsymbol{g}_F$: the ascent gradient $\boldsymbol{g}_f$ of the forgetting samples and the ascent gradient $\boldsymbol{g}_F$ of the other forgetting samples.

It should be noted that $\boldsymbol{g}_F$ is not actually involved in the model update process. Moreover, there are a single pair $\boldsymbol{g}_f$−$\boldsymbol{g}_r$ and a set of pairs $\{\boldsymbol{g}_r - \boldsymbol{g}_{F_k}\}_{k=1}^{|\mathcal{D}_F|}$ and $\{\boldsymbol{g}_f - \boldsymbol{g}_{F_k}\}_{k=1}^{|\mathcal{D}_F|}$ in an update step because we need to consider the gradient of each sample in $\mathcal{D}_F$.

Based on Definition 2, we empirically measure the percentage of data samples having these three direction conflicts during the update of the NegGrad method. As shown in Fig. 1, these observations indicate low conflict for the $\boldsymbol{g}_f$−$\boldsymbol{g}_F$ pair but evident conflict for both $\boldsymbol{g}_r$−$\boldsymbol{g}_F$ and $\boldsymbol{g}_f$−$\boldsymbol{g}_F$ pairs.



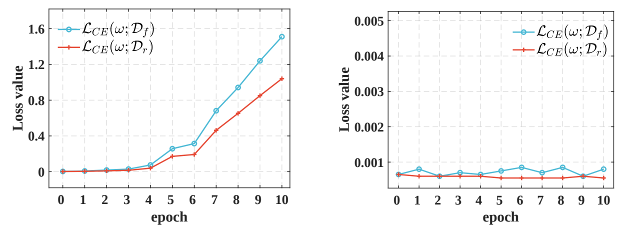
(a) $\boldsymbol{g}_f$−$\boldsymbol{g}_r$ direction conflict (b) $\boldsymbol{g}_r$−$\boldsymbol{g}_F$ direction conflict

Figure 2: Illustrations of the gradient direction problems. g^α denotes the orthogonal gradient component of $\boldsymbol{g}$.

When these direction conflicts occur, we will explain the negative impact on performance with two specific illustrations.

Direction conflict of $\boldsymbol{g}_f$−$\boldsymbol{g}_r$. As shown in Fig. 2(a), direction of the gradient component $\boldsymbol{g}_f^\alpha$ is opposite to $\boldsymbol{g}_r$. As a result, updating the model with $\boldsymbol{g}_f$ will maximize $\mathcal{L}_{\text{CE}}(\omega; \boldsymbol{x}_r, \boldsymbol{y}_r)$ so that the model tends to unlearn the remaining sample $(\boldsymbol{x}_r, \boldsymbol{y}_r)$. Similarly, updating the model with $\boldsymbol{g}_r$ tends to maintain the forgetting data sample $(\boldsymbol{x}_f, \boldsymbol{y}_f)$. These direction conflicts will also cause conflicts between the unlearning task and the maintaining task to decline the performance.

Direction conflict of $\boldsymbol{g}_r$−$\boldsymbol{g}_F$. As shown in Fig. 2(b), $\boldsymbol{g}_r$ and $\boldsymbol{g}_F$ have direction conflict so that updating the model with $\boldsymbol{g}_r$ will also minimize $\mathcal{L}_{\text{CE}}(\omega; \boldsymbol{x}_F, \boldsymbol{y}_F)$, *i.e.,* the model tends to maintain the forgetting data sample $(\boldsymbol{x}_F, \boldsymbol{y}_F)$ rather than unlearn it.



(a) Dominated by $\boldsymbol{g}_f$ (b) Dominated by $\boldsymbol{g}_r$

Figure 3: Measurement of the loss values on CIFAR-10 using ResNet-18 when the gradient dominant problem occurs. Both of the forgetting loss $\mathcal{L}_{\text{CE}}(\omega; \mathcal{D}_f)$ and the remaining loss $\mathcal{L}_{\text{CE}}(\omega; \mathcal{D}_r)$ increase when dominated by $\boldsymbol{g}_f$. Both loss values are maintained almost unchanged when dominated by $\boldsymbol{g}_r$.

3.4 Gradient Domination Problem

The MU process can be seen as multi-task learning, and there are two tasks in the MU process: the forgetting task and the maintaining task. The forgetting task aims to forget the forgetting set, while the maintaining task aims to maintain the remaining set. Ideally, the related gradients of each task jointly update the model to learn the specific knowledge of each task. However, the gradient domination problem is a common challenge that the model's updates during training are biased towards optimizing for one task over others, *i.e.,* a single gradient dominates the model update.

We empirically evaluate this problem through the values of the loss function. As shown in Fig. 3 (a), increasing the remaining loss will decline the model classification performance. On the contrary, the barely changed forgetting loss will lead to invalid forgetting, as shown in Fig. 3 (b). Inspired by the previous work, Yu et al. [36] pointed out that the magnitude of gradients will affect the gradient domination in MTL. Therefore, we try to evaluate the effect of magnitude on this dominant problem by assigning a constant magnitude parameter to the gradients. As shown in Table 1, the results show that the gradient dominant problem always occurs during the NegGrad unlearning process.

Table 1: Effect of the magnitude on the gradient dominant problem by assigning different weight parameters. " – " denotes that the loss values are almost unchanged, and "↑ " denotes that both the loss values are increased. We see that the model update process is dominated by either g_r or g_f, resulting in unchanged or increased loss values.

Magnitude Weight		Dominant	Loss values
g_f	g_r		
0.1	0.9	g_r	–
0.2	0.8	g_r	–
0.3	0.7	g_r	–
0.4	0.6	g_f	↑
0.5	0.5	g_f	↑
0.6	0.4	g_f	↑
0.7	0.3	g_f	↑
0.8	0.2	g_f	↑
0.9	0.1	g_f	↑

4 PROPOSED METHOD

In Sec. 4.1, we propose the Gradient Direction Rectification (GDR) method for the gradient direction problem. In Sec. 4.2, we propose the Gradient Magnitude Adjustment (GMA) method for the domination problem. Finally, we will describe the combined GDR-GMA for the sample-wise MU task in Sec. 4.3.

4.1 Gradient Direction Rectification (GDR)

In Sec. 3.3, we define three gradients that affect the NegGrad MU process, *i.e.,* g_f, g_r, and g_F. However, it is very time-consuming to calculate the set of gradients $\{g_{F_k}\}_{k=1}^{|\mathcal{D}_F|}$. To address this problem, we empirically measure the cosine similarity between gradients of the same sample in adjacent epochs.

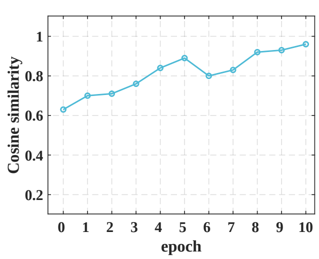
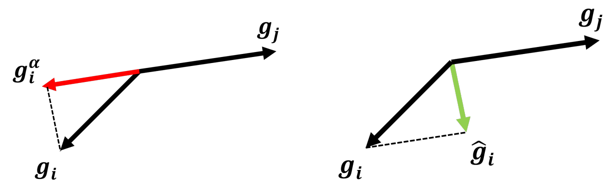


Figure 4: The cosine similarity between gradients in adjacent epochs remains high during the MU process on CIFAR-10 using ResNet-18.

As shown in Fig. 4, this observation indicates that the gradients of the same sample in the adjacent epoch are highly similar during the MU process. Based on this observation, we use a gradient bank to replace the repeated calculation process with storing the gradients in the previous epoch. We formulate this replacement process as:

$$\{g_{B_k}^{e}\}_{k=1}^{|Bank|} := \{g_{F_k}^{e-1}\}_{k=1}^{|\mathcal{D}_F|}, \tag{3}$$



(a) Direction conflict **(b) Gradient projection**

Figure 5: When the direction conflict occurs (a), the GDR method will project the gradient onto the orthonormal plane of the conflicting gradient (b).

where e denotes the current epoch, $e-1$ denotes the previous epoch, and $|Bank|$ is the size of gradient bank, which equals to $|\mathcal{D}_F|$.

We propose the Gradient Direction Rectification (GDR) method to alleviate these direction conflicts by rectifying the direction between the two conflicting gradients g_i and g_j if $\cos\varphi_{ij} < 0$. After rectification of the direction, the gradient i is projected onto the orthonormal plane of the conflicting gradient j to remove destructive conflicts, as shown in Fig. 5. Formally, GDR projects the gradient i onto the orthonormal plane of the gradient j as:

$$\hat{g}_i := g_i - \frac{g_i \cdot g_j}{||g_j||^2} g_j. \tag{4}$$

As described in Sec. 3.3, we analyze the direction conflict problem only exists between the pairs of g_f–g_r and g_f–g_F. However, multiple projections will be needed to alleviate direction conflicts. To avoid the excessive direction deviation caused by the multiple projections, we define the joint conflict gradients in the gradient bank as:

$$g_{joint} = \frac{\sum_{k=1}^{|Bank|} \mathbf{1}(\cos\varphi(g_r, g_{B_k}^{e}) < 0) \cdot g_{B_k}^{e}}{\sum_{k=1}^{|Bank|} \mathbf{1}(\cos\varphi(g_r, g_{B_k}^{e}) < 0)}, \tag{5}$$

where $\mathbf{1}(g_i < 0)$ is an indicator function which yields a value of 1 if $g_i < 0$ and 0 otherwise. Then, GDR can once project g_r onto the plane of the joint gradient instead of the multiple projections.

Algorithm 1: Gradient Direction Rectification (GDR)

for *e in epochs* **do**
  $g_f \leftarrow \nabla_{\omega_t} \mathcal{L}_{\text{CE}}(\omega_t; x_f, y_f)$
  $g_r \leftarrow \nabla_{\omega_t} \mathcal{L}_{\text{CE}}(\omega_t; x_r, y_r)$
  $g_{joint} \leftarrow \frac{\sum_{k=1}^{|Bank|} \mathbf{1}(\cos\varphi(g_r, g_{B_k}^{e})<0) \cdot g_{B_k}^{t}}{\sum_{k=1}^{|Bank|} \mathbf{1}(\cos\varphi(g_r, g_{B_k}^{e})<0)}$
  if $\cos\varphi(g_f, g_r) < 0)$ **then**
    $\hat{g}_f \leftarrow g_f - \frac{g_f \cdot g_r}{||g_r||^2} g_r$
    $\hat{g}_r \leftarrow g_r - \frac{g_r \cdot g_f}{||g_f||^2} g_f$
  end
  if $\cos\varphi(\hat{g}_r, g_{joint}) < 0)$ **then**
    $\hat{g}_r \leftarrow \hat{g}_r - \frac{\hat{g}_r \cdot g_{joint}}{||g_{joint}||^2} g_{joint}$
  end
end

We present the GDR method in Algorithm 1. The proposed algorithm utilize three types of gradients g_f, g_r, and g_F. We use an approximate joint gradient $\boldsymbol{g}_{joint}$ from the gradient bank to reduce computational overhead and avoid multiple projections. If conflicts are detected, we project the gradient based on Eq. 4 to alleviate direction conflicts.

4.2 Gradient Magnitude Adjustment (GMA)

In Sec. 3.4, we empirically observe that the magnitude will affect the gradient domination. However, simply adjusting the magnitude cannot balance the two gradients and instead results in domination by a single gradient, as shown in Table 1. Therefore, we propose a Gradient Magnitude Adjustment (GMA) method to dynamically adjust the magnitude. Specifically, GMA can dynamically adjust the magnitude of update gradients according to the three states of the MU process as follows:

- ***State I (Maintaining Dominance):*** In this state, the remaining loss value maintains nearly the same as that in the original model, which shows that the remaining gradient $\boldsymbol{g}_r$ dominates the MU process. Hence, GMA will assign a large magnitude to the forgetting gradient $\boldsymbol{g}_f$ to balance the forgetting and maintaining tasks.
- ***State II (Balance):*** In this state, none of the dominance occurs. Hence, GMA needs to flexibly adjust the magnitude of the update gradients to maintain this balance.
- ***State III (Forgetting Dominance):*** When the remaining loss dramatically deviates from the original loss value, GMA will assign a large magnitude to the remaining gradient $\boldsymbol{g}_r$ to force the model to maintain the remaining samples.

Formally, we define the dynamic magnitude weight (DMW) as:

$$\lambda_t = \frac{1}{1 + \exp(\gamma \cdot (\Delta\ell_t - \epsilon))}, \tag{6}$$

where t denotes the t-th unlearning step, ϵ is a constant small value, γ denotes the steepness parameter, and the loss deviation $\Delta\ell_t = |\mathcal{L}_{CE}(\boldsymbol{\omega}_u^t; \boldsymbol{x}_r^t) - \mathcal{L}_{CE}(\boldsymbol{\omega}_0; \mathcal{D}_r)|$ denotes the difference between the loss values of the remaining samples in the t-th step and the remaining dataset in the original model. The steepness parameter γ controls how sharply the weight parameter transitions from its minimum to maximum value. A large ϵ will tolerate a larger increase in the loss values of the remaining samples. With the DMW parameter λ_t, GMA assigns it to gradients in the t-th update step as:

$$\boldsymbol{\omega}_u^{t+1} \leftarrow \boldsymbol{\omega}_u^t + \eta[\lambda_t \boldsymbol{g}_f^t - (1 - \lambda_t)\boldsymbol{g}_r^t]. \tag{7}$$

Furthermore, we will describe why the above proposed DMW parameter can suit these three states. As shown in Fig. 6, we illustrate the relationship between the DMW parameter λ_t and the loss deviation $\Delta\ell_t$ in the three states. In State I, GMA adjusts λ_t to be close to 1 so that the forgetting loss will rapidly ascend to the balance state. A low decay rate is used in this state to make λ_t insensitive to the loss deviation and close to 1. In State II, when the loss deviation $\Delta\ell_t$ is around the small value ϵ, GMA flexibly adjusts the magnitude of $\boldsymbol{g}_f$ and $\boldsymbol{g}_r$ with a high change rate of λ_t to keep the balance between forgetting and maintaining tasks. In State III, when the loss deviation $\Delta\ell_t$ is greater than the small value ϵ, GMA adjusts the magnitude of $\boldsymbol{g}_r$ to close to 1 with a low change rate of λ_t. With a large magnitude of $\boldsymbol{g}_r$, GMA can make the model tend to maintain the remaining samples and get back to the balance state as soon as possible.

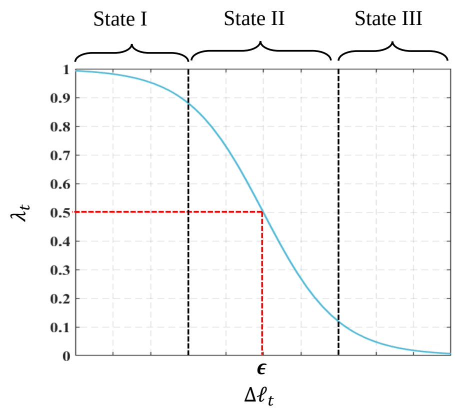


Figure 6: Illustration of the relationship between the dynamic weight parameter λ_t and the loss deviation $\Delta\ell_t$ in the three states. *State I*: $\Delta\ell_t$ is close to 0 and λ_t tends to be 1 with a low decay rate; *State II*: $\Delta\ell_t$ is around the preset small value ϵ and λ_t is sensitive to $\Delta\ell_t$ a high decay rate; *State III*: $\Delta\ell_t$ is much higher than ϵ and λ_t will be close to 0 with a low decay rate.

Implementing GMA is simple and barely brings additional computational overhead because it only requires numerical loss values. Through the crafted DMW parameter, GMA can dynamically adjust the update gradients' magnitude to balance these two tasks.

4.3 Machine Unlearning via GDR-GMA

In Sec. 4.1, we propose the GDR method to rectify the direction of two pairs of conflicting gradients. Despite alleviating the direction conflicts, GDR cannot address the gradient dominant problem. Therefore, in Sec. 4.2, we propose the GMA method to dynamically adjust the magnitude of the update gradients to balance the forgetting task and the remaining task.

Algorithm 2: GDR-GMA Unlearning Process

```
for e in epochs do
    for t in steps do
        // Get the rectified gradients by GDR
        ĝ_f, ĝ_r ← GDR
        // Get the dynamic weight by GMA
        λ_t ← GMA
        // Update the model
        ω_u^{t+1} ← ω_u^t + η[λ_t ĝ_f^t − (1 − λ_t) ĝ_r^t].
    end
end
```

We present the combined GDR-GMA method in Algorithm 2. First, GDR-GMA rectifies the direction of the conflicting gradients by the GDR method. Then, GDR-GMA calculates the dynamic weight parameter using the GMA method. Finally, GDR-GMA updates the model with the rectified gradients and dynamic magnitude weight parameters.

Table 2: Performance comparison with several baselines in 20% random data forgetting and 50% random data forgetting. The performance gap against Retrain is provided in (Δ). The optimal Avg. Gap is marked in red.

Dataset & Model	Approach	Random Data Forgetting (20%)						Random Data Forgetting (50%)					
		$Acc_{\mathcal{D}_f}$ (Δ ↓)	$Acc_{\mathcal{D}_r}$ (Δ ↓)	Acc_{val} (Δ ↓)	MIA (Δ ↓)	Avg. Gap ↓	RTE ↓	$Acc_{\mathcal{D}_f}$ (Δ ↓)	$Acc_{\mathcal{D}_r}$ (Δ ↓)	Acc_{val} (Δ ↓)	MIA (Δ ↓)	Avg. Gap ↓	RTE ↓
CIFAR-10 ResNet-18	Retrain	94.00$_{\pm0.54}$ (0.00)	100.00$_{\pm0.00}$ (0.00)	94.00$_{\pm0.12}$ (0.00)	76.76$_{\pm0.04}$ (0.00)	0.00	28.53	91.73$_{\pm0.15}$ (0.00)	100.00$_{\pm0.00}$ (0.00)	91.99$_{\pm0.12}$ (0.00)	70.79$_{\pm0.03}$ (0.00)	0.00	19.71
	Fine-tune	85.82$_{\pm1.12}$ (8.18)	88.07$_{\pm0.24}$ (11.93)	86.98$_{\pm0.87}$ (7.02)	32.09$_{\pm0.14}$(44.67)	17.95	1.48	87.48$_{\pm1.52}$ (4.25)	88.98$_{\pm0.44}$ (11.02)	86.78$_{\pm0.26}$ (5.21)	66.81$_{\pm0.31}$(3.98)	6.12	1.04
	Random Labels	81.58$_{\pm1.12}$ (12.42)	85.67$_{\pm1.06}$ (14.33)	83.60$_{\pm0.21}$ (10.40)	59.69$_{\pm0.25}$(17.07)	13.56	1.21	81.45$_{\pm0.66}$ (10.28)	84.51$_{\pm0.56}$ (15.49)	82.44$_{\pm0.12}$ (9.55)	45.35$_{\pm0.29}$ (25.44)	15.19	2.19
	NegGrad	72.81$_{\pm0.89}$ (21.19)	76.58$_{\pm1.61}$ (23.42)	71.48$_{\pm1.64}$ (22.52)	76.25$_{\pm0.89}$ (0.51)	16.91	1.33	72.69$_{\pm1.57}$ (19.04)	75.70$_{\pm1.49}$ (24.30)	70.50$_{\pm1.12}$ (21.49)	76.54$_{\pm1.23}$ (5.75)	17.64	2.07
	Fisher [9]	23.28$_{\pm2.56}$(70.72)	23.96$_{\pm3.41}$(76.04)	23.10$_{\pm3.27}$(70.90)	62.23$_{\pm0.28}$(14.53)	58.05	117.82	22.39$_{\pm1.45}$(69.34)	22.63$_{\pm2.15}$(77.37)	22.47$_{\pm2.31}$(69.52)	62.64$_{\pm0.53}$(8.15)	61.10	76.01
	Unrolling [33]	92.97$_{\pm0.55}$(1.03)	93.72$_{\pm0.38}$(6.28)	87.58$_{\pm0.11}$(6.42)	44.30$_{\pm0.24}$(32.46)	11.55	0.27	85.14$_{\pm0.36}$(6.59)	89.73$_{\pm0.23}$(10.27)	85.34$_{\pm0.21}$(6.65)	43.66 $_{\pm0.14}$(27.13)	12.66	0.44
	BadT [5]	87.12$_{\pm0.15}$(6.88)	91.74$_{\pm0.39}$(8.26)	87.83$_{\pm0.16}$(6.17)	47.81$_{\pm0.14}$(28.95)	12.57	1.12	85.93$_{\pm0.28}$(5.80)	89.48$_{\pm0.32}$(10.52)	85.78$_{\pm0.20}$(6.21)	27.35$_{\pm0.29}$(43.44)	16.50	1.14
	SSD [8]	97.72$_{\pm0.72}$ (3.72)	97.51$_{\pm0.04}$ (2.49)	91.73$_{\pm0.46}$ (2.27)	88.04$_{\pm0.46}$ (11.28)	4.94	3.04	99.99$_{\pm0.01}$ (8.26)	100.00$_{\pm0.00}$ (0.00)	94.98$_{\pm0.25}$ (2.99)	92.04$_{\pm0.88}$ (21.25)	8.12	3.17
	L_1-Sparse [24]	92.72$_{\pm0.62}$(1.28)	96.81$_{\pm0.30}$(3.19)	91.62$_{\pm0.58}$(2.38)	71.28$_{\pm0.04}$(5.48)	3.08	2.25	90.81$_{\pm0.81}$(0.92)	94.11$_{\pm0.26}$(5.89)	88.37$_{\pm0.73}$(3.62)	65.75$_{\pm0.22}$(5.04)	3.87	1.39
	SalUn [7]	93.87$_{\pm0.66}$ (0.13)	99.17$_{\pm0.21}$ (0.83)	92.53$_{\pm0.54}$ (1.47)	68.47$_{\pm0.98}$ (8.29)	2.68	2.77	92.46$_{\pm0.59}$ (0.73)	98.06$_{\pm0.51}$ (1.94)	89.93$_{\pm0.06}$ (2.06)	62.88$_{\pm0.81}$ (7.91)	3.16	1.67
	GDR-GMA (ours)	93.84$_{\pm0.04}$ (0.16)	99.22$_{\pm0.11}$ (0.78)	92.40$_{\pm1.47}$ (1.60)	82.36$_{\pm0.06}$ (5.60)	**2.04**	1.40	92.10$_{\pm0.08}$ (0.63)	99.07$_{\pm0.45}$ (0.93)	90.37$_{\pm1.10}$ (1.62)	76.27$_{\pm0.18}$ (5.48)	**2.17**	2.12
Tiny-ImageNet ViT	Retrain	52.87$_{\pm0.62}$ (0.00)	90.29$_{\pm0.71}$ (0.00)	54.18 $_{\pm0.70}$ (0.00)	47.27$_{\pm0.08}$ (0.00)	0.00	846.71	47.75$_{\pm0.43}$ (0.00)	92.46$_{\pm0.52}$ (0.00)	48.54 $_{\pm0.37}$ (0.00)	44.83$_{\pm0.22}$ (0.00)	0.00	602.18
	Fine-tune	70.85$_{\pm0.93}$ (17.98)	88.30$_{\pm0.85}$ (1.99)	52.71$_{\pm0.99}$ (1.47)	53.81$_{\pm0.39}$ (6.54)	7.00	27.04	70.75$_{\pm0.98}$ (23.00)	92.33$_{\pm0.50}$ (0.13)	51.76$_{\pm0.91}$ (3.22)	51.70$_{\pm0.49}$ (6.87)	8.30	17.74
	Random Labels	75.76$_{\pm0.76}$ (22.89)	82.71$_{\pm0.31}$ (7.58)	52.83$_{\pm0.75}$ (1.35)	27.87$_{\pm0.28}$ (19.40)	12.81	16.86	79.99$_{\pm0.88}$ (32.24)	82.33$_{\pm0.53}$ (10.13)	52.83$_{\pm0.76}$ (4.29)	34.35$_{\pm0.65}$ (10.48)	14.28	41.66
	NegGrad	84.58$_{\pm1.19}$ (31.71)	85.56$_{\pm0.58}$ (4.73)	51.52$_{\pm1.26}$ (2.66)	63.91$_{\pm0.34}$ (16.64)	13.93	13.17	51.19$_{\pm0.66}$ (3.44)	51.79$_{\pm1.47}$ (40.67)	36.34$_{\pm1.28}$ (12.20)	53.62$_{\pm1.21}$ (8.79)	16.27	23.57
	Unrolling [33]	92.66$_{\pm0.68}$ (39.79)	92.79$_{\pm0.39}$ (2.50)	54.47$_{\pm0.71}$ (0.29)	68.37$_{\pm0.54}$ (21.10)	15.92	7.83	89.06$_{\pm0.64}$ (41.31)	89.18$_{\pm0.46}$ (3.28)	52.60$_{\pm0.71}$ (4.06)	66.07$_{\pm0.52}$ (21.24)	17.47	19.52
	BadT [5]	48.20$_{\pm0.49}$ (4.67)	57.73$_{\pm0.30}$ (32.56)	43.24$_{\pm0.47}$ (10.94)	29.45$_{\pm0.25}$ (17.82)	16.50	10.32	45.78$_{\pm0.51}$ (1.97)	54.35$_{\pm0.27}$ (38.11)	41.91$_{\pm0.56}$ (6.63)	26.30$_{\pm0.34}$ (18.53)	16.31	10.43
	SSD [8]	92.97$_{\pm0.77}$ (40.10)	92.75$_{\pm0.24}$ (2.46)	54.35$_{\pm0.68}$ (0.17)	68.50$_{\pm0.26}$ (21.23)	15.99	25.64	92.80$_{\pm0.85}$ (45.05)	92.83$_{\pm0.28}$ (0.37)	54.35$_{\pm0.79}$ (5.81)	68.25$_{\pm0.21}$ (23.42)	18.66	27.70
	L_1-Sparse [24]	68.71$_{\pm0.57}$ (15.84)	85.13$_{\pm0.21}$ (5.16)	50.25$_{\pm0.50}$ (3.93)	44.26$_{\pm0.32}$ (3.01)	6.99	32.15	69.04$_{\pm1.47}$ (21.29)	87.55$_{\pm0.78}$ (4.91)	49.78$_{\pm1.06}$ (1.24)	43.32$_{\pm0.70}$ (1.51)	7.24	20.17
	SalUn [7]	54.99$_{\pm0.43}$ (2.12)	89.27$_{\pm0.29}$ (1.02)	52.35$_{\pm0.40}$ (1.83)	40.27$_{\pm0.24}$ (7.00)	2.99	34.83	52.21$_{\pm0.47}$ (4.46)	89.65$_{\pm0.40}$ (2.81)	50.45$_{\pm0.48}$ (1.91)	38.21$_{\pm0.60}$ (6.62)	3.95	24.38
	GDR-GMA (ours)	51.69$_{\pm0.48}$ (1.18)	91.49$_{\pm0.02}$ (1.20)	51.78$_{\pm0.06}$ (2.40)	52.91$_{\pm0.16}$ (5.64)	**2.61**	13.67	46.25$_{\pm0.28}$ (1.50)	90.45$_{\pm0.23}$ (2.01)	45.06$_{\pm0.60}$ (3.48)	50.92$_{\pm0.22}$ (6.09)	**3.27**	25.60

5 EXPERIMENTS

In this Section, we conduct extensive experiments to empirically evaluate the proposed GDR-GMA method. We compare its performance with several baseline MU methods in three MU scenarios: **random data forgetting**, **subclass forgetting**, and **class forgetting**. Furthermore, we also conduct ablation experiments to prove the effectiveness of GDR-GMA.

5.1 Experiment Setups

Datasets and models. In this paper, we evaluate GDR-GMA on three datasets, CIFAR-10, CIFAR-20 [20], and Tiny-ImageNet [21], using two model architectures, ResNet-18 [13] and Vision Transformer (ViT) [6]. More details of the datasets can be found in the supplementary materials.

Baseline methods. In random data forgetting and subclass forgetting, we compare GDR-GMA with the following baselines: **Retrain**, **Fine-tune**, **Random Labels** [9], **NegGrad**, **Fisher** [9], **Unrolling** [33], **BadT** [5], **L_1-Sparse** [24], **SalUn** [7], **SSD** [32]. Furthermore, we add two new baselines in class forgetting, *i.e.,* **ERM-KTP** [23] and **Boundary** [3]. More setups of implementation are presented in supplementary materials.

Metrics. Following previous works [5, 7, 32], we use $Acc_{\mathcal{D}_f}$, $Acc_{\mathcal{D}_r}$, and Acc_{val} to represent the classification accuracy on the forgetting set, the remaining set, and the validation set, respectively. Besides, we leverage the membership inference attack (MIA) on the forgetting set to measure whether the forgetting data samples are in the training set. We use RTE to measure the time overhead in minutes of the unlearning process. We use $Acc_{\mathcal{D}_f}$ and MIA to measure the effectiveness of MU methods, $Acc_{\mathcal{D}_r}$ and Acc_{val} to evaluate fidelity, and RTE to measure efficiency. To present a summary performance gap against the ideal baseline Retrain, we introduce the Avg. Gap by calculating the average performance gaps in $Acc_{\mathcal{D}_f}$, $Acc_{\mathcal{D}_r}$, Acc_{val} and MIA. Note that **the better performance** of an MU method corresponds to **the smaller performance gap** with **Retrain**. The results are given by a format $a_{\pm b}$ with mean a and standard deviation b over ten independent experiments.

Hyper-parameters. The original models are trained for 200 epochs using the 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. For our proposed GDR-GMA, we set the steepness parameter γ to 100 and the constant small value ϵ to 0.02.

5.2 Evaluation of the GDR-GMA Method

Comparison experiments in random data forgetting. We conduct extensive comparison experiments with several baselines. First, we evaluate the performance of forgetting a random subset of data samples. Following previous works [5, 7, 32], we consider two unlearning scenarios, *i.e.,* 20% random data forgetting and 50% random data forgetting. More results can be found in the supplementary materials. Based on the results presented in Table 2, we draw the following three key observations:

First, following the previous work [7], Avg. Gap is a more comprehensive metric to evaluate the performance of the MU methods. Some methods may be the strongest when considering only a single metric, but this comes at the cost of sacrificing the other metrics. However, GDR-GMA still achieves the smallest average performance gap against Retrain on these two (data-model) setups in both scenarios, demonstrating its superior effectiveness and fidelity.

Second, GDR-GMA inherits the efficiency of the gradient ascent methods while maintaining the model classification performance. GDR-GMA significantly improves the performance of NegGrad with negligible additional computation overhead. Furthermore, GDR-GMA has a competitive computation efficiency with these baselines, as evidenced by the RTE metric.

Third, randomly forgetting 50% data samples is a more complex scenario, resulting in a higher average performance gap than that in 20% random data forgetting scenario. Besides, the MU methods on Tiny-ImageNet using ViT are more demanding due to the larger scale of data samples and model parameters. Nevertheless, GDR-GMA still has the smallest average performance gap and achieves superior performance on both effectiveness and fidelity.

Comparison experiments in subclass forgetting. We then explore the performance of forgetting a subclass of data samples, following previous works [5, 8]. Sub-class forgetting is a simpler MU scenario than random data forgetting because sub-class samples are more similar. This simple data distribution makes the model more easily unlearn the forgetting data samples while maintaining the remaining data samples. As shown in Table 3, GDR-GMA achieves a superior performance in sub-class forgetting than random data forgetting. Moreover, GDR-GMA has the smallest performance gap and significantly outperforms the state-of-the-art works regarding time overhead.

Table 3: Performance comparison with several baselines on CIFAR-20 using ResNet-18 in subclass forgetting.

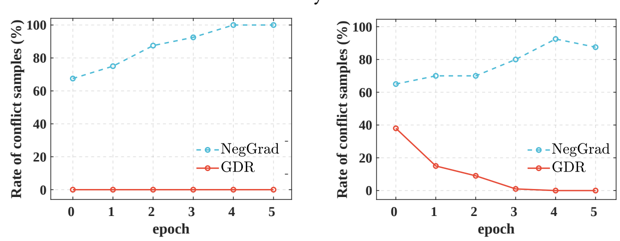
Approach	$Acc_{\mathcal{D}_f}(\Delta\downarrow)$	$Acc_{\mathcal{D}_r}$ $(\Delta\downarrow)$	Acc_{val} $(\Delta\downarrow)$	MIA $(\Delta\downarrow)$	Avg. Gap ↓	RTE ↓
Retrain	$2.52_{\pm0.62}$ (0.00)	$100.00_{\pm0.00}$ (0.00)	$68.94_{\pm0.62}$ (0.00)	$0.00_{\pm0.00}$(0.00)	0.00	35.74
Fine-tune	$3.33_{\pm0.14}$ (0.81)	$90.64_{\pm0.21}$ (9.36)	$68.22_{\pm0.19}$ (0.72)	$0.00_{\pm0.00}$ (0.00)	2.72	1.79
Random Labels	$4.80_{\pm0.17}$ (2.28)	$69.97_{\pm0.38}$ (30.03)	$62.13_{\pm0.23}$ (6.81)	$0.00_{\pm0.00}$ (0.00)	9.78	1.13
NegGrad	$31.30_{\pm0.15}$ (28.78)	$97.57_{\pm0.37}$ (2.43)	$80.04_{\pm0.02}$ (11.10)	$0.09_{\pm0.01}$ (0.09)	10.60	0.52
Unrolling [33]	$15.07_{\pm0.04}$ (12.55)	$97.30_{\pm0.13}$ (2.70)	$83.15_{\pm0.52}$ (14.21)	$0.31_{\pm0.02}$ (0.31)	7.44	0.43
BadT [5]	$7.77_{\pm0.20}$ (5.25)	$94.81_{\pm0.74}$ (5.19)	$72.15_{\pm0.07}$ (3.21)	$0.10_{\pm0.01}$ (0.10)	3.44	1.85
SSD [8]	$0.10_{\pm0.00}$ (2.42)	$70.45_{\pm0.48}$ (29.55)	$59.78_{\pm0.18}$ (9.16)	$32.31_{\pm0.28}$ (32.31)	18.36	3.25
L_1-Sparse [24]	$3.70_{\pm0.11}$ (1.18)	$86.58_{\pm0.25}$ (13.42)	$67.99_{\pm0.23}$ (0.95)	$0.00_{\pm0.00}$ (0.00)	3.89	1.94
SalUn [7]	$2.68_{\pm0.96}$ (0.16)	$96.21_{\pm0.60}$ (3.79)	$66.89_{\pm0.06}$ (2.05)	$0.00_{\pm0.00}$ (0.00)	1.50	2.16
GDR-GMA (ours)	$2.01_{\pm0.09}$ (0.51)	$99.76_{\pm0.02}$ (0.24)	$67.99_{\pm0.36}$ (0.95)	$0.00_{\pm0.00}$ (0.00)	**0.42**	0.60

Comparison experiments in class forgetting. We also conduct experiments in class forgetting, following the previous works [3, 23, 32]. Except for the baseline methods in random data forgetting and sub-class forgetting, we add two class-wise MU methods as baselines, ERM-KTP [23] and Boundary [3]. As shown in Table 4, GDR-GMA still has the smallest average performance gap and has less time overhead than the SOTA works.

Table 4: Performance comparison with several baselines on CIFAR-10 using ResNet-18 in class forgetting.

Approach	$Acc_{\mathcal{D}_f}(\Delta\downarrow)$	$Acc_{\mathcal{D}_r}$ $(\Delta\downarrow)$	Acc_{val} $(\Delta\downarrow)$	MIA $(\Delta\downarrow)$	Avg. Gap ↓	RTE ↓
Retrain	$0.00_{\pm0.00}$ (0.00)	$100.00_{\pm0.00}$ (0.00)	$79.64_{\pm0.62}$ (0.00)	$7.30_{\pm0.00}$(0.00)	0.00	33.20
Fine-tune	$0.00_{\pm0.00}$ (0.00)	$96.47_{\pm0.54}$ (3.53)	$78.43_{\pm0.61}$ (1.21)	$24.22_{\pm0.45}$ (16.92)	5.41	1.81
Random Labels	$0.00_{\pm0.00}$ (0.00)	$86.97_{\pm0.78}$ (13.03)	$84.70_{\pm0.53}$ (5.06)	$0.28_{\pm0.04}$ (7.02)	6.28	0.87
NegGrad	$19.36_{\pm0.60}$ (19.36)	$97.66_{\pm0.96}$ (2.34)	$84.29_{\pm0.84}$ (4.65)	$22.14_{\pm0.98}$ (14.84)	10.30	0.71
Unrolling [33]	$1.56_{\pm0.21}$ (1.56)	$84.10_{\pm0.86}$ (15.90)	$78.69_{\pm0.43}$ (0.95)	$16.28_{\pm0.09}$ (8.98)	6.85	0.31
BadT [5]	$0.00_{\pm0.00}$ (0.00)	$94.94_{\pm0.54}$ (5.06)	$80.79_{\pm0.91}$ (1.15)	$0.00_{\pm0.00}$ (7.30)	3.38	1.41
SSD [8]	$0.00_{\pm0.00}$ (0.00)	$94.65_{\pm0.14}$ (5.35)	$92.55_{\pm0.80}$ (12.91)	$0.00_{\pm0.00}$ (7.30)	6.39	3.10
L_1-Sparse [24]	$0.00_{\pm0.00}$ (0.00)	$90.07_{\pm0.14}$ (9.93)	$79.70_{\pm0.88}$ (0.06)	$13.92_{\pm0.23}$ (6.62)	4.15	1.65
SalUn [7]	$0.00_{\pm0.00}$ (0.00)	$95.89_{\pm0.19}$ (4.11)	$82.76_{\pm0.28}$ (3.12)	$0.67_{\pm0.27}$ (6.63)	3.47	2.84
ERM-KTP [23]	$0.00_{\pm0.00}$ (0.00)	$96.62_{\pm0.66}$ (3.38)	$78.10_{\pm0.82}$ (1.54)	$34.27_{\pm0.45}$ (26.97)	7.97	2.02
Boudary [3]	$4.67_{\pm0.11}$ (4.67)	$98.84_{\pm0.97}$ (1.16)	$78.23_{\pm0.69}$ (1.41)	$0.00_{\pm0.00}$ (7.30)	3.63	1.83
GDR-GMA (ours)	$0.00_{\pm0.00}$ (0.00)	$96.26_{\pm0.75}$ (3.74)	$81.76_{\pm0.46}$ (2.12)	$4.00_{\pm0.27}$ (3.30)	**2.29**	0.75

The experimental results in these three scenarios demonstrate the superior effectiveness, fidelity, and efficiency of GDR-GMA. Furthermore, the simple implementation and adaptability across scenarios also show the scalability of GDR-GMA.



(a) Direction conflict of g_f−g_r (b) Direction conflict of g_r−g_F

Figure 7: Evaluation of the effectiveness of the GDR method on CIFAR-10 using ResNet-18. After GDR rectifies the direction of the conflicting gradients, the rate of conflicting samples is decreased to 0%.

Effectiveness of GDR. We evaluate the effectiveness of the GDR method in handling the problem of direction conflict. As shown in Fig.7, GDR can significantly alleviate the direction conflict problem. However, Fig. 7(b) shows that the rate of conflict samples is not 0% in the first few epochs due to the approximation error using the gradient bank. Specifically, the approximation error is caused because the cosine similarity between adjacent epochs is around 0.6-0.8, as shown in Fig. 4. Nevertheless, the rate reaches 0% after a few epochs because the error will be negligible when the cosine similarity increases during the unlearning process.

Effectiveness of GMA. Then, we evaluate the effectiveness of the GDR method in handling the gradient dominant problem. As shown in Fig. 8, these two tasks can achieve a balance where each gradient fulfills its duty instead of solely tending to one of them after the GMA method dynamically adjusts the magnitude of the update gradients. Furthermore, it demonstrates that GMA can achieve a balance between the updated gradients.

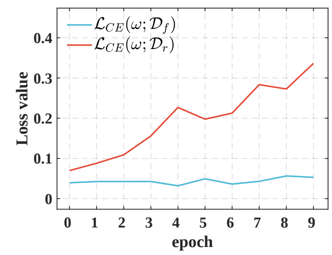


Figure 8: Evaluation of the effectiveness of the GMA method on CIFAR-10 using ResNet-18. After GMA adjusts the magnitude of the update gradients, the forgetting loss value increases with maintaining the remaining loss.

5.3 Ablation Experiments

In this section, we conduct ablation experiments to further analyze the effectiveness of GDR-GMA, and more ablation experiments can be found in supplementary materials.

Table 5: Ablation experiments of the each pair of gradients on CIFAR-10 using ResNet-18.

g_f-g_r	g_r-g_F	g_f-g_F	$Acc_{\mathcal{D}_f}(\Delta\downarrow)$	$Acc_{\mathcal{D}_r}$ $(\Delta\downarrow)$	Acc_{val} $(\Delta\downarrow)$	MIA $(\Delta\downarrow)$	Avg. Gap ↓
✗	✗	✗	$90.21_{\pm0.06}$ (3.79)	$96.21_{\pm0.13}$ (3.77)	$90.65_{\pm1.67}$ (3.35)	$85.79_{\pm0.07}$ (9.03)	4.89
✓	✗	✗	$93.68_{\pm0.05}$ (0.32)	$98.55_{\pm0.11}$ (1.45)	$91.88_{\pm1.50}$ (2.12)	$83.11_{\pm0.06}$ (6.35)	2.56
✗	✓	✗	$93.29_{\pm0.07}$ (0.71)	$98.30_{\pm0.12}$ (1.70)	$91.62_{\pm1.66}$ (2.38)	$83.49_{\pm0.08}$ (6.73)	2.88
✗	✗	✓	$90.18_{\pm0.06}$ (3.82)	$96.12_{\pm0.13}$ (3.98)	$90.61_{\pm1.67}$ (3.39)	$85.74_{\pm0.07}$ (8.98)	5.04
✓	✗	✓	$93.64_{\pm0.05}$ (0.36)	$98.54_{\pm0.11}$ (1.46)	$91.86_{\pm1.50}$ (2.14)	$83.08_{\pm0.06}$ (6.32)	2.57
✗	✓	✓	$93.31_{\pm0.07}$ (0.69)	$98.24_{\pm0.12}$ (1.76)	$91.60_{\pm1.66}$ (2.36)	$83.52_{\pm0.08}$ (6.76)	2.89
✓	✓	✓	$93.83_{\pm0.04}$ (0.17)	$99.18_{\pm0.11}$ (0.82)	$92.35_{\pm1.47}$ (1.65)	$82.40_{\pm0.06}$ (5.64)	2.07
✓	✓	✗	$93.84_{\pm0.04}$ (0.16)	$99.22_{\pm0.11}$ (0.78)	$92.40_{\pm1.47}$ (1.60)	$82.36_{\pm0.06}$ (5.60)	**2.04**

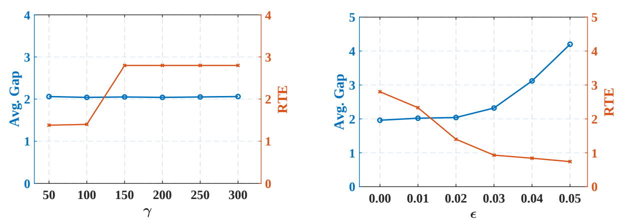
Impact of the each pair of gradients. In Sec. 3.3, we define three pairs of gradients and empirically find two conflicting pairs $\boldsymbol{g}_f-\boldsymbol{g}_r$ and $\boldsymbol{g}_r-\boldsymbol{g}_F$. As shown in Table 5, the average performance gap is improved by handling the direction conflicts of $\boldsymbol{g}_f-\boldsymbol{g}_r$ and $\boldsymbol{g}_r-\boldsymbol{g}_F$. Besides, handling the direction conflict of $\boldsymbol{g}_f-\boldsymbol{g}_F$ does not achieve a smaller performance gap due to the low rate of conflict samples, as our observations in Fig. 1.

Table 6: Ablation experiments of the proposed gradient bank on CIFAR-10 using ResNet-18.

	Random Data Forgetting (20%)					
	$Acc_{\mathcal{D}_f}(\Delta\downarrow)$	$Acc_{\mathcal{D}_r}$ $(\Delta\downarrow)$	Acc_{val} $(\Delta\downarrow)$	MIA $(\Delta\downarrow)$	Avg. Gap ↓	RTE ↓
w/o Bank	$93.86_{\pm0.05}$ (0.14)	$99.23_{\pm0.11}$ (0.77)	$92.43_{\pm1.50}$ (1.57)	$82.32_{\pm0.05}$ (5.56)	2.01	102.32
w/ Bank	$93.84_{\pm0.04}$ (0.16)	$99.22_{\pm0.11}$ (0.78)	$92.40_{\pm1.47}$ (1.60)	$82.36_{\pm0.06}$ (5.60)	2.04	**1.40**
	Random Data Forgetting (50%)					
	$Acc_{\mathcal{D}_f}(\Delta\downarrow)$	$Acc_{\mathcal{D}_r}$ $(\Delta\downarrow)$	Acc_{val} $(\Delta\downarrow)$	MIA $(\Delta\downarrow)$	Avg. Gap ↓	RTE ↓
w/o Bank	$92.06_{\pm0.11}$ (0.67)	$99.24_{\pm0.40}$ (0.76)	$90.40_{\pm1.32}$ (1.59)	$77.02_{\pm0.12}$ (6.23)	2.31	122.42
w/ Bank	$92.10_{\pm0.08}$ (0.63)	$99.07_{\pm0.45}$ (0.93)	$90.37_{\pm1.10}$ (1.62)	$76.27_{\pm0.18}$ (5.48)	2.17	**2.12**

Impact of the gradient bank. In Sec. 4.1, we propose the gradient bank to approximate the set of gradients $\{\boldsymbol{g}_{F_k}\}_{k=1}^{|\mathcal{D}_F|}$ with $\boldsymbol{g}_{joint}$. Fig. 7(b) demonstrates the effectiveness of this approximation using the gradient bank to alleviate the direction conflict. Furthermore, the gradient bank can significantly reduce time overhead without sacrificing performance, as shown in Table 6.

Impact of γ and ϵ. We propose the dynamic magnitude weight parameter in Eq. 6 to adjust the magnitude of the update gradients, resulting in a balance between the forgetting and remaining tasks. Specifically, we use the steepness parameter γ to control how sharply the weight parameter transitions from its minimum value to its maximum value and the constant small value ϵ to control the deviation of the loss value of the remaining dataset. As shown in Fig. 7, the results show that the value of γ does not affect the average performance gap but affects the time overhead. The higher value of ϵ will reduce the time overhead with an increasing average performance gap.



(a) The steepness parameter γ **(b) The constant small value ϵ**

Figure 9: Ablation experiments of the steepness parameter γ (a) and the constant small value ϵ (b) in Eq. 6.

Evaluation of GDR and GMA. The GDR-GMA MU process consists of two methods, *i,e,* GDR to rectify the direction and GMA to adjust the magnitude. These two methods mainly focus on the gradient direction problem and gradient dominant problem, respectively. Here, we want to explore the relationship between problems and performance. In Sec. 3, we have observed that the occurrences of problems indeed significantly decline the performance. As shown in Table 7, we evaluate the performance when only considering a single problem. The results demonstrate that GDR-GMA achieves the best performance by handling both problems.

Table 7: Ablation experiments to evaluate the performance of GDR and GMA on CIFAR-10 using ResNet-18.

Method	$Acc_{\mathcal{D}_f}$(Δ ↓)	$Acc_{\mathcal{D}_r}$ (Δ ↓)	Acc_{val} (Δ ↓)	MIA (Δ ↓)	Avg. Gap ↓	RTE ↓
GDR	$91.34_{\pm 0.05}$ (2.66)	$90.10_{\pm 0.12}$ (9.89)	$88.26_{\pm 1.76}$ (5.74)	$81.05_{\pm 0.04}$ (4.29)	5.65	1.40
GMA	$90.21_{\pm 0.06}$ (3.79)	$96.21_{\pm 0.13}$ (3.77)	$90.65_{\pm 1.67}$ (3.35)	$85.79_{\pm 0.07}$ (9.03)	4.89	1.38
GDR-GMA	$93.84_{\pm 0.04}$ (0.16)	$99.22_{\pm 0.11}$ (0.78)	$92.40_{\pm 1.47}$ (1.60)	$82.36_{\pm 0.06}$ (5.60)	**2.04**	1.40

6 DISCUSSIONS

In this section, we want to propose some insights of GDR-GMA. Specifically, we will discuss that GDR-GMA can easily combine with the existing methods to further improve the performance. In addition, we will discuss the possibility of applying GDR-GMA to other computer vision tasks.

Combining with the Existing Methods. Our method only modifies the direction and magnitude of update gradients rather than changing the model architecture and the pipeline of the gradient ascent. Therefore, our proposed GDR-GMA can be easily combined with the existing methods as long as they do not conflict with the basic gradient ascent process. For example, Liu et al. [24] proposed that the model sparsity can improve the MU performance. Hence, the L_1-Sparse method can be easily combined with our proposed GDR-GMA. We modify the model update formulation by adding the gradient of the L_1 norm penalty term as:

$$\omega_u^{t+1} \leftarrow \omega_u^t + \eta[\lambda_t \boldsymbol{g}_f^t - (1-\lambda_t)\boldsymbol{g}_r^t - \mu \boldsymbol{g}_{L_1}^t], \tag{8}$$

where $\boldsymbol{g}_{L_1} = \nabla_\omega ||\omega||_1$ and μ is a regularization parameter. Then, we conduct experiments to evaluate the performance of the combined method. As shown in Table 8, we can observe that the combined L_1-Sparse + GDR-GMA achieves a smaller performance gap than GDR-GMA with a higher time overhead. We analyze the reasons and find that the sparsity forces the model only to update the filtered important parameters, which slows down the increase of the forgetting loss value. As a result, the loss value can be closer to Retrain but brings more additional time overhead. More results can be found in the supplementary materials.

Table 8: Evaluation of combining GDR-GMA with L_1-Sparse on CIFAR-10 using ResNet-18.

	Random Data Forgetting (20%)					
Method	$Acc_{\mathcal{D}_f}$(Δ ↓)	$Acc_{\mathcal{D}_r}$ (Δ ↓)	Acc_{val} (Δ ↓)	MIA (Δ ↓)	Avg. Gap ↓	RTE ↓
L_1-Sparse [24]	$92.72_{\pm 0.62}$(1.28)	$96.81_{\pm 0.30}$(3.19)	$91.62_{\pm 0.58}$(2.38)	$71.28_{\pm 0.04}$(5.48)	3.08	2.25
GDR-GMA	$93.84_{\pm 0.04}$ (0.16)	$99.22_{\pm 0.11}$ (0.78)	$92.40_{\pm 1.47}$ (1.60)	$82.36_{\pm 0.06}$ (5.60)	2.04	1.40
L_1-Sparse + GDR-GMA	$93.26_{\pm 0.08}$ (0.74)	$99.68_{\pm 0.25}$ (0.32)	$92.62_{\pm 2.09}$ (1.38)	$81.23_{\pm 0.18}$ (4.47)	1.73	2.96
	Random Data Forgetting (50%)					
Method	$Acc_{\mathcal{D}_f}$(Δ ↓)	$Acc_{\mathcal{D}_r}$ (Δ ↓)	Acc_{val} (Δ ↓)	MIA (Δ ↓)	Avg. Gap ↓	RTE ↓
L_1-Sparse [24]	$90.81_{\pm 0.81}$(0.92)	$94.11_{\pm 0.26}$(5.89)	$88.37_{\pm 0.73}$(2.06)	$65.75_{\pm 0.22}$(5.04)	3.48	1.39
GDR-GMA	$92.10_{\pm 0.08}$ (0.63)	$99.07_{\pm 0.45}$ (0.93)	$90.37_{\pm 1.10}$ (1.62)	$76.27_{\pm 0.18}$ (5.48)	2.17	2.12
L_1-Sparse + GDR-GMA	$92.23_{\pm 0.19}$ (0.50)	$99.54_{\pm 1.12}$ (0.46)	$89.22_{\pm 1.97}$ (2.77)	$74.56_{\pm 0.45}$ (3.77)	1.89	4.06

GDR-GMA for the semantic segmentation task. We design the GDR-GMA method based on the gradient ascent, which can theoretically be applied to other computer vision tasks. Therefore, we attempt to apply the GDR-GMA for the semantic segmentation task with a toy example on the OxfordPet dataset [29] using FPN architecture [19]. To implement GDR-GMA for the semantic segmentation task, we only need to replace gradients calculated on CE loss with gradients on dice loss. As shown in Table 9, GDR-GMA achieves a small performance gap against Retrain, demonstrating the high scalability potential of GDR-GMA. For other machine learning tasks, we will leave them for future work. The details of setups and more results can be found in the supplement materials.

Table 9: Performance of GDR-GMA for semantic segmentation on OxfordPet using FPN in 20% random data forgetting.

	Random Data Forgetting (20%)					
Method	$Acc_{\mathcal{D}_f}$(Δ ↓)	$Acc_{\mathcal{D}_r}$ (Δ ↓)	Acc_{val} (Δ ↓)	MIA (Δ ↓)	Avg. Gap ↓	RTE ↓
Retrain	$90.28_{\pm 0.57}$ (0.00)	$94.59_{\pm 0.14}$ (0.00)	$89.95_{\pm 0.48}$ (0.00)	$49.61_{\pm 0.85}$ (0.00)	0.00	1.63
Fine-tune	$92.50_{\pm 0.12}$ (2.22)	$93.66_{\pm 0.41}$ (0.93)	$90.13_{\pm 0.52}$ (0.18)	$52.54_{\pm 0.47}$ (2.93)	1.56	0.29
Random Labels	$90.13_{\pm 0.08}$ (0.15)	$92.42_{\pm 0.07}$ (2.17)	$88.83_{\pm 0.91}$ (1.12)	$42.90_{\pm 0.84}$ (6.71)	2.54	0.19
NegGrad	$89.22_{\pm 0.99}$ (1.06)	$91.45_{\pm 0.25}$ (3.14)	$87.82_{\pm 0.40}$ (2.13)	$59.29_{\pm 0.43}$ (9.68)	4.00	0.19
Unrolling [33]	$92.58_{\pm 0.20}$ (2.30)	$92.96_{\pm 0.19}$ (1.63)	$89.51_{\pm 0.20}$ (0.44)	$50.73_{\pm 0.69}$ (1.12)	1.37	0.19
BadT [5]	$89.53_{\pm 0.37}$ (0.75)	$90.03_{\pm 0.36}$ (4.56)	$87.13_{\pm 0.44}$ (2.82)	$51.91_{\pm 0.88}$ (2.30)	2.61	0.63
L_1-Sparse [24]	$91.65_{\pm 0.76}$ (1.37)	$92.33_{\pm 0.40}$ (2.26)	$89.67_{\pm 0.32}$ (0.28)	$58.50_{\pm 0.20}$ (8.89)	3.20	0.39
SalUn [7]	$93.33_{\pm 0.42}$ (3.05)	$93.66_{\pm 0.63}$ (0.93)	$90.30_{\pm 0.70}$ (0.35)	$55.41_{\pm 0.41}$ (5.80)	2.53	0.20
GDR-GMA	$89.77_{\pm 0.39}$ (0.51)	$92.86_{\pm 0.82}$ (1.73)	$89.43_{\pm 0.57}$ (0.52)	$50.88_{\pm 0.43}$ (1.27)	**1.01**	0.20

7 CONCLUSION

In this paper, we first analyze the problems of the basic gradient ascent MU method in a multi-task learning view. We identify two key problems: the gradient direction problem and the gradient dominant problem. To address these problems, we propose the GDR-GMA MU method. For the gradient direction problem, GDR-GMA rectifies the direction of the conflicting gradients. For the gradient dominant problem, GDR-GMA dynamically adjusts the magnitude of the update gradients. Then, we conducted extensive experiments to demonstrate the effectiveness, fidelity, and efficiency of GDR-GMA. Furthermore, we also show the scalability of GDR-GMA by exploring the possibility of combining the existing methods and applying GDR-GMA to the semantic segmentation task.

REFERENCES

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

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*. Springer, 213–229.

[3] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. 2023. Boundary Unlearning: Rapid Forgetting of Deep Networks via Shifting the Decision Boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7766–7775.

[4] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*. PMLR, 794–803.

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

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[7] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. 2023. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508* (2023).

[8] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. 2023. Fast machine unlearning without retraining through selective synaptic dampening. *arXiv preprint arXiv:2308.07707* (2023).

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

[10] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11516–11524.

[11] Anisa Halimi, Swanand Kadhe, Ambrish Rawat, and Nathalie Baracaldo. 2022. Federated unlearning: How to efficiently erase a client in fl? *arXiv preprint arXiv:2207.05521* (2022).

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[14] Tuan Hoang, Santu Rana, Sunil Gupta, and Svetha Venkatesh. 2024. Learn to Unlearn for Deep Neural Networks: Minimizing Unlearning Interference with Gradient Projection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 4819–4828.

[15] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)* 54, 11s (2022), 1–37.

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[17] Mostafa Kahla, Si Chen, Hoang Anh Just, and Ruoxi Jia. 2022. Label-only model inversion attacks via boundary repulsion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15045–15053.

[18] Mahdi Khosravy, Kazuaki Nakamura, Yuki Hirose, Naoko Nitta, and Noboru Babaguchi. 2022. Model inversion attack by integration of deep generative models: Privacy-sensitive face generation from a face recognition system. *IEEE Transactions on Information Forensics and Security* 17 (2022), 357–372.

[19] Alexander Kirillov, Kaiming He, Ross Girshick, and Piotr Dollár. 2017. A unified architecture for instance and semantic segmentation. In *Computer Vision and Pattern Recognition Conference*. CVPR.

[20] A Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Tront* (2009).

[21] Ya Le and Xuan Yang. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7, 7 (2015), 3.

[22] Guanghao Li, Li Shen, Yan Sun, Yue Hu, Han Hu, and Dacheng Tao. 2023. Subspace based Federated Unlearning. *arXiv preprint arXiv:2302.12448* (2023).

[23] Shen Lin, Xiaoyu Zhang, Chenyang Chen, Xiaofeng Chen, and Willy Susilo. 2023. ERM-KTP: Knowledge-Level Machine Unlearning via Knowledge Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20147–20155.

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

[25] Shikun Liu, Edward Johns, and Andrew J Davison. 2019. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1871–1880.

[26] James Martens. 2020. New insights and perspectives on the natural gradient method. *The Journal of Machine Learning Research* 21, 1 (2020), 5776–5851.

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

[28] Ngoc-Bao Nguyen, Keshigeyan Chandrasegaran, Milad Abdollahzadeh, and Ngai-Man Cheung. 2023. Re-thinking Model Inversion Attacks Against Deep Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16384–16393.

[29] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3498–3505.

[30] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).

[31] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

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

[33] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. 2022. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 303–319.

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

[35] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. 2022. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM Web Conference 2022*. 622–632.

[36] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems* 33 (2020), 5824–5836.