

Robust Machine Unlearning for Quantized Neural Networks via Adaptive Gradient Reweighting with Similar Labels

Yujia Tong¹, Yuze Wang¹, Jingling Yuan^{1*}, Chuang Hu²

¹ Wuhan University of Technology, ² University of Macau

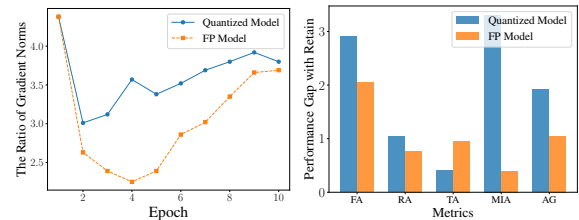
{tyjjjj, hbtmwy, yjl}@whut.edu.cn, chuanguhu@um.edu.mo

Abstract

Model quantization enables efficient deployment of deep neural networks on edge devices through low-bit parameter representation, yet raises critical challenges for implementing machine unlearning (MU) under data privacy regulations. Existing MU methods designed for full-precision models fail to address two fundamental limitations in quantized networks: 1) Noise amplification from label mismatch during data processing, and 2) Gradient imbalance between forgotten and retained data during training. These issues are exacerbated by quantized models' constrained parameter space and discrete optimization. We propose *Q-MUL*, the first dedicated unlearning framework for quantized models. Our method introduces two key innovations: 1) *Similar Labels* assignment replaces random labels with semantically consistent alternatives to minimize noise injection, and 2) *Adaptive Gradient Reweighting* dynamically aligns parameter update contributions from forgotten and retained data. Through systematic analysis of quantized model vulnerabilities, we establish theoretical foundations for these mechanisms. Extensive evaluations on benchmark datasets demonstrate *Q-MUL*'s superiority over existing approaches.

1. Introduction

Deep neural networks (DNNs) have achieved significant success in fields such as computer vision [14] and natural language processing [28], but their high demand for computation and memory limits their deployment on resource-constrained edge devices. Model quantization [9, 22, 32] reduces the storage requirements and computational complexity by lowering the bit-width of model weights and activations, making them more suitable for edge devices. To enhance performance, user-provided data can be utilized for periodic online fine-tuning of quantized models on edge devices. However, with the implementation of data privacy



(a) Comparison of gradient. (b) Comparison of performance.

Figure 1. We conduct Random Labels [11] on the full-precision and quantized models. (a) The ratio of gradient norms between the forgotten data and the retained data in the loss. (b) Performance gaps of FA, RA, TA, and MIA with Retrain, AG is the average gap. The observations are from ResNet18 on CIFAR-10.

regulations such as GDPR [15], users have the right to request the deletion of their uploaded data, which involves not only removing data from databases but also ensuring that specific samples no longer influence the trained models. This raises a new challenge: **How to eliminate the influence of specific data samples on quantized models?**

Machine unlearning (MU) provides a potential pathway to implement the *right to be forgotten* through parametric influence removal. Existing MU approaches fall into two categories: 1) *Exact unlearning* [3] achieves ideal performance by retraining the model from scratch after removing the data to be forgotten, but it requires a high computational cost. 2) *Approximate unlearning* [5] achieves comparable unlearning performance by adjusting the parameters of a pre-trained model, requiring fewer computational resources. *We focus on approximate unlearning due to its speed and efficiency making it particularly well-suited for deployment on edge devices.*

Existing MU methods face two main limitations when applied to quantized models since they [8, 11] are primarily designed for full-precision models and do not adequately address the impact of quantization. Consequently, when these methods (e.g., Random Labels [11] and SalUn [8]) are directly applied to quantized models, their effectiveness is reduced, particularly in two key stages: data processing

*Corresponding author

and unlearning training. During the data processing phase, existing methods [8, 11] simulate the forgetting process by assigning random labels to samples in the forgotten dataset. However, the introduction of random labels introduces substantial noise that is inconsistent with the original data. Using these noisy samples in subsequent unlearning training can cause significant deviations in the direction of model parameter updates, thereby impairing the model’s robustness. This issue is exacerbated in quantized models, as their limited parameter representation space makes them more sensitive to noise than full-precision models.

Furthermore, in the unlearning training phase, we observed a substantial discrepancy in the gradient norms when training with forgotten versus retained data. This discrepancy results in uneven contributions from the two datasets to the model’s parameter updates, thereby reducing the effectiveness of unlearning. While this issue is present in full-precision models, it is more pronounced in quantized models. As shown in Figure 1, the y-axis represents the ratio of gradient norms when using forgotten data to the gradient norms when using retained data. After unlearning, the average performance gap of quantized models compared to Re-train is 1.92%, which is significantly higher than the 1.05% gap observed in full-precision models (a smaller gap indicates a more effective forgetting process). This suggests that current methods lead to a degradation in the effectiveness of unlearning when applied to quantized models.

To address these challenges, we propose **Q-MUL**—a quantization-aware unlearning framework with dual innovations. During the data processing phase, we select the label closest to the true label as the label (that we term Similar Labels) for the samples in the forgotten dataset. During the unlearning training process, we adaptively weight the loss of the forgotten data and the retained data to balance their contributions to the updates of model parameters. We have conducted extensive experiments to validate the effectiveness of Q-MUL, and when applied to quantized models, its performance is superior to existing methods. We summarize our contributions below:

- To the best of our knowledge, we are the first to apply unlearning techniques directly to quantized models. We analyze the challenges involved, including the noise introduced during the data processing phase and the excessive gradient discrepancy between forgotten and retained data during unlearning training.
- We propose Q-MUL, a method that uses Similar Labels to replace the labels of forgotten data samples during the data processing phase, thereby reducing noise. In the unlearning training phase, we adaptively adjust the loss weights for forgotten and retained data based on the gradients, ensuring a balanced contribution to the model parameter updates.
- We conduct extensive experiments on benchmark datasets

to evaluate the performance of Q-MUL. The results demonstrate that Q-MUL outperforms existing MU methods, making it more suitable for quantized models.

2. Related Work

Our work bridges two distinct research domains: *machine unlearning* for privacy compliance and *model quantization* for efficient deployment. We analyze their intersection and identify critical gaps in existing literature.

Machine Unlearning has evolved from theoretical frameworks to practical implementations. The seminal work by Ginart et al. [10] formalized unlearning requirements for k-means clustering, while Bourtole et al. [3] introduced the “SISA” framework for deep neural networks through sharded training. Recent advances focus on approximate unlearning paradigms: Sekhari et al. [25] established theoretical guarantees for gradient-based updates, and Golatkar et al. [11] proposed noise injection in parameter space for influence removal. However, these methods assume continuous parameter spaces (32-bit FP) and unrestricted gradient dynamics – assumptions invalidated by quantized models’ discrete optimization landscapes. Notably, existing MU techniques exhibit three limitations in quantized contexts: 1) Random label assignment [8] introduces catastrophic noise in low-bit regimes, 2) Uniform gradient treatment [11] fails to address the amplified imbalance caused by quantized representations, and 3) No existing work considers the interaction between backpropagation through quantizers and machine unlearning.

Model Quantization [4, 30, 31] is a key method in deep learning model compression technology, which reduces the size and computational requirements of the model by lowering the bit width of model parameters. Model quantization research has progressed along two main paths: Post-Training Quantization (PTQ) [22] that adjusts pre-trained models, and Quantization-Aware Training (QAT) [7] that embeds quantization during learning. PTQ directly quantizes the model after training without additional training, but it can lead to a significant degradation in model performance. QAT simulates quantization operations during the model training process, allowing the model to adapt to the errors introduced by quantization from the outset, which can alleviate the decline in model performance. PACT [6] employs a learnable clipping threshold optimized during training for low-precision quantization. DSQ [12] introduces a differentiable soft tanh function to approximate standard quantization steps. As the most widely used and representative QAT method, LSQ+ [2] uses learnable quantization parameters to achieve better accuracy. Since MU methods typically require training with data, unlearning training can be viewed as a process of Quantization-Aware Training. This paper is committed to proposing an unlearning approach suitable for Quantization-Aware Training.

3. Preliminaries

In this section, we first revisit the concept of *machine unlearning*. Subsequently, we introduce the fundamental computational process of *Quantization-Aware Training*.

3.1. Revisiting Machine Unlearning

Let the complete (training) dataset $D = \{(x_i, y_i)\}_{i=1}^N$ be with N number of samples, in which $x_i \in \mathbb{R}^d$ is the i^{th} sample with the associated class label $y_i \in \{1, 2, \dots, K\}$. The forgotten dataset $D_f \subseteq D$ represents a subset of the complete dataset D , typically denoting the data that requires deletion from the trained model. And we denote its complement as retained dataset D_r , which represents the data we want to retain, *i.e.*, $D_f \cap D_r = \emptyset$, $D_f \cup D_r = D$. Based on the composition of D_f , machine unlearning (MU) in image classification can be divided into two categories: class-wise forgetting and random data forgetting. In class-wise forgetting, the data in D_f consists entirely of samples from the same class, with the purpose of eliminating the influence of an entire class on the model. In random data forgetting, the data in D_f may include random samples from one or multiple classes, with the purpose of eliminating the influence of randomly selected data points from D_f on the model. Prior to unlearning, we denote the $\mathcal{M}_0$ as the original model. In the context of MU, we denote retrained model $\mathcal{M}_r$ as the model trained from scratch solely on the retained dataset, which is regarded as the “golden standard” [24]. Nonetheless, retraining entails a substantial computational overhead. Therefore, approximate unlearning was proposed, with the primary goal of obtaining unlearned model $\mathcal{M}_u$ by eliminating the influence of forgotten dataset D_f from the original model $\mathcal{M}_0$, thereby approximating retrained model $\mathcal{M}_r$ with less computational overhead.

3.2. Quantization-Aware Training

Quantization-Aware Training (QAT) employs fake quantization nodes to simulate quantization error during inference. These nodes perform quantization and dequantization operations on floating-point numbers. For n -bit signed quantization, given the scale factor s , we can define a quantization function $q(\cdot)$ to represent the operations within the fake quantization node:

$$x^q = q(x^r) = s \times \left\lfloor \text{clamp}\left(\frac{x^r}{s}, -2^{n-1}, 2^{n-1} - 1\right) \right\rfloor, \quad (1)$$

where $\lfloor \cdot \rfloor$ rounds to the nearest integer, and $\text{clamp}(x, r_{\text{low}}, r_{\text{high}})$ ensures x is within $[r_{\text{low}}, r_{\text{high}}]$. In neural networks with quantized activations and weights, forward and backward propagation can be summarized as:

Forward: $\text{Output}(x) = x^q \cdot w^q = q(x^r) \cdot q(w^r)$,

$$\text{Backward: } \frac{\partial \mathcal{L}}{\partial x^r} = \begin{cases} \frac{\partial \mathcal{L}}{\partial x^q} & \text{if } x \in [-Q_N^x, Q_P^x] \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

$$\frac{\partial \mathcal{L}}{\partial w^r} = \begin{cases} \frac{\partial \mathcal{L}}{\partial w^q} & \text{if } w \in [-Q_N^w, Q_P^w] \\ 0 & \text{otherwise} \end{cases},$$

where $\mathcal{L}$ is the loss function, $q(\cdot)$ is used during forward propagation, and the straight-through estimator (STE) [1] is employed for gradient derivation during backward propagation. In image classification tasks, neural networks map input images x to predicted probability distributions $p(x; w)$ via the forward function $f(x; w)$. For any sample (x_i, y_i) in the dataset $D = \{(x_i, y_i)\}_{i=1}^N$, the quantized model’s output probability distribution (logits) is:

$$p_Q(x_i^q; w^q) = f(x_i^q; w^q), \quad (3)$$

In order to simplify the notation, we will use x_i to denote the input and w to denote the quantized model weights in the following text.

4. Methodology

In this section, we provide a comprehensive introduction to the proposed Q-MUL. We first introduce the overview of Q-MUL. Subsequently, we provide a detailed description of two components of Q-MUL: *Similar Labels* and *Adaptive Gradient Reweighting*.

4.1. The Overview of Q-MUL

In Figure 2, we present an overview of Q-MUL, which is divided into two stages: data processing and unlearning training. During the data processing stage, Q-MUL selects the label closest to the true label (**Similar Labels**) as the training label for the samples in the forgotten dataset D_f , while keeping the labels of the samples in the retained dataset D_r unchanged, resulting in the processed datasets D'_f and D'_r for the next stage of training. In the unlearning training stage, for each epoch, Q-MUL first calculates the gradient norms of the loss function on datasets D'_f and D'_r , and then adaptively calculates their weights based on the gradient norms and incorporates them into the loss function. After several epochs of training, an unlearned model is obtained. In the next two subsections of this section, we will provide a detailed introduction to the two stages of Q-MUL.

4.2. Similar Labels

In MU, to achieve the forgetting of specific samples (samples in the forgotten dataset), it is common to train by randomly shuffling the labels of the samples in the forgotten dataset [8, 11]. However, this method has some obvious shortcomings. Random labels (RL) do not take into account

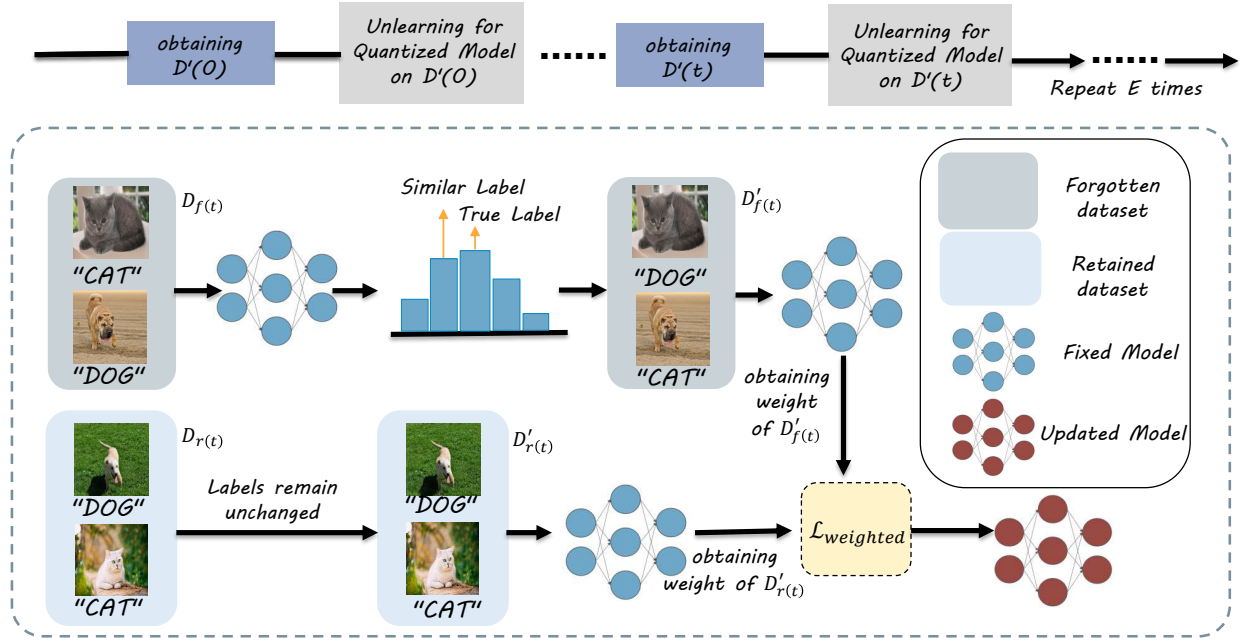


Figure 2. The overview of Q-MUL. Before performing the unlearning process on the model, Q-MUL first selects the label that is closest to the true label in the current output of the model as the new label for the data samples that need to be forgotten. During the unlearning process, we adaptively calculate the weights of the forgotten and retained data based on their gradient norms.

the potential relationships between data, introducing a large amount of noise inconsistent with the original data. Assume that the true label of the data to be forgotten is “cat”. If we use a random label, such as “car”, which is clearly unrelated to the original data, this may lead to a significant difference in the direction of parameter updates when the model is trained on these data compared to using the original label, thereby affecting the overall stability and Robustness of the model, causing it to forget information that should not be forgotten.

To address these issues, we propose Similar Labels (SL). SL selects labels that are closest to the original label, thereby achieving the forgetting of specific data while reducing the negative impact on model performance. This method helps to maintain the consistency and logic of the data because it reduces the amount of noise introduced and can better maintain the model’s predictive ability for similar samples. Moreover, SL can more precisely control the forgetting process because it aligns the direction of model parameter updates (gradient direction) with the true labels, thereby avoiding the introduction of substantial random noise. This helps to mitigate the potential over-adjustment of the model during the unlearning process.

Specifically, we first input the forgetting data into the model. For a forgetting instance $(x_i, y_i) \in D_f$, the model’s predicted probability distribution is $p_Q(x_i; w)$. The proba-

bility predicted by the model corresponding to the true label is $p_{gt} = p_Q(y_i | x_i; w)$. For each class $k \in \{1, 2, \dots, K\}$ of the model output, we calculate the distance between its predicted probability and the true label probability $d(k)$:

$$d(k) = |p_Q(k | x_i; w) - p_{gt}|, \quad (4)$$

Then, we find the similar label k_{sl} that minimizes the distance $d(k)$:

$$k_{sl} = \arg \min_{k \neq y_i} d(k). \quad (5)$$

Finally, we use these similar labels k_{sl} instead of the original labels y_i . To better understand the selection of Similar Labels, we provide the pseudocode of the selection process in Algorithm 1.

We conduct a theoretical analysis by comparing the differences in gradient update directions between SL and RL. Let the model parameters be w , the original label of sample x_i be y_i , the similar label be k_{sl} , and the random label be k_{rl} . During the training process, the cross-entropy loss can be expressed as:

$$\mathcal{L} = - \sum_{(x_i, y_i) \in D} \sum_{k=1}^K y_{i,k} \log p_Q(k | x_i; w) \quad (6)$$

where $y_{i,k}$ is the one-hot encoding of the true label y_i , i.e., $y_{i,k} = 1$ if and only if $k = y_i$, otherwise $y_{i,k} = 0$. There-

Algorithm 1 Similar Labels

Input: Original complete dataset $D = \{(x_i, y_i)\}_{i=1}^N$, The quantized model $\mathcal{M}_t$ with parameters w_t at epoch t

Output: Training dataset D' for unlearning

```
1: initialize  $D'_f = \emptyset, D'_r = \emptyset$ ;  
2: for  $i \in [1, \dots, N]$  do  
3:   if  $(x_i, y_i) \in D_f$  then  
4:     Calculate predicted probability  $p_Q(x_i; w)$  by eq.(3);  
5:     Calculate distance  $d(k)$  following eq.(4);  
6:     Calculate class label  $k_{sl}$  by eq.(5);  
7:     Add  $(x_i, k_{sl})$  to  $D'_f$ ;  
8:   else  
9:     Add  $(x_i, y_i)$  to  $D'_r$ ;  
10:  end if  
11: end for  
12:  $D' = D'_r \cup D'_f$ ;  
13: return  $D'$ 
```

fore, the loss function can be simplified as follows:

$$\mathcal{L} = - \sum_{(x_i, y_i) \in D} \log p_Q(y_i | x_i; w) \quad (7)$$

where $p_Q(y_i | x_i; w)$ is the probability that the model predicts the true class y_i for the sample x_i . For the sample (x_i, y_i) , the gradient of the loss function is:

$$\nabla_w \mathcal{L} = -\nabla_w \log p_Q(y_i | x_i; w) \quad (8)$$

We measure the difference in gradient directions using cosine similarity. The cosine of the angle θ_{sl} between the gradients when using SL and the original label is:

$$\cos \theta_{sl} = \frac{\nabla_w \log p_Q(y_i | x_i; w) \cdot \nabla_w \log p_Q(k_{sl} | x_i; w)}{\|\nabla_w \log p_Q(y_i | x_i; w)\| \cdot \|\nabla_w \log p_Q(k_{sl} | x_i; w)\|} \quad (9)$$

The cosine of the angle θ_{rl} between the gradients when using RL and the original label has an approximate form. Since the selection of k_{sl} makes $p_Q(k_{sl} | x_i; w)$ close to p_{gt} , the angle θ_{sl} between the gradient directions of SL and the true label is small, hence $\cos \theta_{sl} \approx 1$. This indicates that the gradient update direction of similar labels is highly consistent with the gradient direction of the true label, avoiding the gradient conflicts that may arise from random labels. From the perspective of gradient update directions, this ensures the robustness of quantized models after unlearning.

4.3. Adaptive Gradient Reweighting

In methods based on RL [8, 11], the model is trained using the forgotten data with labels randomly shuffled and the retained data with labels unchanged. However, these methods are designed for full-precision models, and their performance significantly degrades when applied to quantized models. Our experiments reveal that the degradation

is due to the large gradient differences between the forgotten and retained data during the unlearning process, leading to inconsistent contributions to model parameter updates. As shown in Figure 1, we apply the RL method to both full-precision and 4-bit quantized models for unlearning and compared the gradient norms of model parameters when trained with forgetting and retained data in each epoch (the y-axis represents the gradient norm when using forgotten data divided by the gradient norm when using retained data). Due to the limited parameter representation range, the differences in gradient norms when training with forgotten and retained data are larger in quantized models than in full-precision models, leading to inconsistent contributions to model parameter updates. Therefore, the performance of these methods degrades when applied to quantized models.

To address this issue, we propose Adaptive Gradient Reweighting (AGR). AGR adaptively adjusts the contributions of forgotten data and retained data to model updates to achieve balance. Specifically, we use a weighted cross-entropy loss function to adjust sample weights:

$$\mathcal{L}_{\text{weighted}}(x, y; w) = \frac{1}{|D'|} \sum_{(x_i, y_i) \in D'} \alpha_i \cdot \mathcal{L}_{\text{CE}}(x_i, y_i; w), \quad (10)$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss function used to calculate the loss of each sample, D' is the training dataset for unlearning, y is the true label and α_i are the weights of each sample used to adjust the contribution of each sample to the total loss. To achieve adaptive adjustment, we design a gradient norm balancing strategy. Before each training epoch, the expected gradient norms of the two types of data are calculated separately:

$$G_f = \mathbb{E}_{(x, y) \sim D'_f} [\|\nabla_w \mathcal{L}_{\text{forget}}(x, y; w)\|_2] \quad (11)$$

$$G_r = \mathbb{E}_{(x, y) \sim D'_r} [\|\nabla_w \mathcal{L}_{\text{retain}}(x, y; w)\|_2] \quad (12)$$

where $\mathcal{L}_{\text{forget}}$ and $\mathcal{L}_{\text{retain}}$ are the loss functions for the forgotten and retained data, respectively. The weight coefficients are dynamically allocated based on the gradient norms:

$$\alpha_f = \frac{G_r}{G_f + G_r}, \quad \alpha_r = \frac{G_f}{G_f + G_r} \quad (13)$$

During the forgetting training process, when using Stochastic Gradient Descent (SGD) to update the model parameters, the parameter update at time t can be expressed as:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t (\alpha_f \nabla \mathcal{L}_{\text{forget}} + \alpha_r \nabla \mathcal{L}_{\text{retain}}) \quad (14)$$

In SGD, model parameters are updated based on the gradient direction, while the gradient magnitude (norm) determines the update step size. If the gradient magnitudes of different data points vary significantly, the model may overemphasize certain data points while neglecting others, leading to instability. Therefore, the weighted gradient

Algorithm 2 The overall pipeline of Q-MUL

Input: The original quantized model $\mathcal{M}_o$ with parameters w_0 , Original complete dataset $D = \{(x_i, y_i)\}_{i=1}^N$, Total training epochs T

Output: The unlearned quantized model $\mathcal{M}_u$ with parameters w_u

- 1: **for** $t \in [0, \dots, T - 1]$ **do**
 - 2: Process the original dataset D to obtain the training dataset D' following Algorithm 1;
 - 3: Calculate gradient norms G_f by eq.(11);
 - 4: Calculate gradient norms G_r by eq.(12);
 - 5: Calculate weight of D'_r and D'_f by eq.(13);
 - 6: Update the parameters w_t of quantized model $\mathcal{M}_t$ by eq.(14);
 - 7: **end for**
 - 8: **return** Unlearned quantized model $\mathcal{M}_u$ with parameters w_u
-

$\nabla \mathcal{L}_{\text{weighted}}$ is obtained by combining the gradients of the forgotten and retained data with their respective weights. If the gradient norm of the forgotten data is greater than that of the retained data, then the weight for the forgotten data will be relatively smaller, and vice versa. This complementarity ensures a more balanced contribution of both types of data in the model parameter updates, thereby enhancing the robustness of the quantized model after unlearning.

To help better understand the unlearning process, we provide the overall pipeline of Q-MUL in Algorithm 2.

5. Experiments

5.1. Experimental Setup

Datasets and Networks. We choose CIFAR-10 [19], CIFAR-100 [19], SVHN [23] and Tiny-Imagenet [20] as the datasets. We evaluate our method on two mainstream deep neural networks, MobileNetV2 [16] and ResNet-18 [14]. We quantize the weights of MobileNetV2 to 2 bits, while keeping the activations in full-precision, and quantize both the weights and activations of ResNet-18 to 4 bits.

Baselines. To demonstrate the superiority of Q-MUL, we choose different types of baselines, which can be classified as: **Retrain**: Using the retained dataset to retrain a model from scratch is the most effective method for machine unlearning, yet it consumes a significant amount of time. **Fine-Tuning (FT)** [11, 29]: Using the retained dataset to slightly fine-tune the original model, unlike retraining, FT requires only a few training epochs. **Gradient Ascent (GA)** [13, 27]: Contrary to gradient descent, GA trains on the forget dataset and updates the parameters of the original model in the direction of gradient ascent. **Influence Unlearning (IU)** [17, 18]: Using influence functions to estimate the impact of D_f on the $\mathcal{M}_0$, and then applying a Newton-one-step-update to the parameters, eliminating the influence of specific data points. **Random Labels (RL)** [11]:

Train on the complete dataset after assigning random labels to the targets in the forgotten dataset. **ℓ_1 -sparse** [21]: Infuse weight sparsity into approximate unlearning through model pruning. **SalUn** [8]: Integrate the gradient-based weight saliency map with the RL method.

Evaluation Metrics. Consistent with ℓ_1 -sparse [21] and SalUn [8], we select multiple evaluation metrics, as follows: **Forget Accuracy (FA)**, which evaluates the model’s accuracy on the forget dataset after undergoing machine unlearning. **Retain Accuracy (RA)**, which evaluates the model’s accuracy on the retained dataset after undergoing machine unlearning. **Test Accuracy (TA)**, which measures the model’s accuracy on the testing dataset, which can be used to assess the generalization ability of the model after undergoing MU. **Membership Inference Attack (MIA)** [26], which is an attack method aimed at determining whether a particular data sample was used to train a given machine learning model. MIA can be used to assess whether a model still retains residual information of certain data after attempting to forget them. Please note that the above four metrics are not simply the higher the better or the lower the better, but rather, they should have as little gap as possible compared to the **Retrain**. Finally, we introduce the **Average Gap (AG)**, which calculates the average gaps between the baseline method and Retrain in terms of the four metrics: FA, RA, TA, and MIA after unlearning. The closer the AG is to 0, the better the unlearning effect.

Implementation details. We use the LSQ+ [2] quantization method unless specified. The original model is trained from scratch for 182 epochs using SGD with a cosine-annealed learning rate (initial 0.1). The Retrain method follows the same optimizer configuration but reduces training to 160 epochs. We conduct experiments under two unlearning scenarios: random data forgetting and class-wise forgetting, with a primary focus on random data forgetting due to its prevalence in real-world applications. Other MU training details and additional experimental results are provided in the Appendix.

5.2. Performance Evaluation

As presented in Table 1, we demonstrate the performance of various MU methods for quantized ResNet18 under random data forgetting scenario. Compared to all the baseline methods, the performance of Q-MUL is the best. Specifically, On CIFAR-10, existing methods have already achieved relatively small average gaps, but Q-MUL is still able to further reduce the gap compared to Retrain. On CIFAR-100, when the proportion of forgotten data samples to all samples is 10%, the FA and RA of Q-MUL are 75.71% and 97.89% respectively (with gaps of 0.95% and 2.09% from Retrain), while the previous state-of-the-art (SOTA) method SalUn has FA and RA of 82.22% and 98.71% respectively (with gaps of 7.46% and 1.27% from Retrain). This indi-

Method	CIFAR-10					CIFAR-100				
	FA	RA	TA	MIA	AG↓	FA	RA	TA	MIA	AG↓
The proportion of forgotten data samples to all samples is 10%										
Retrain	93.38	100.0	92.64	13.36	0	74.76	99.98	72.43	56.36	0
FT	99.42(6.04)	99.96(0.04)	93.05(0.41)	2.82(10.54)	4.26	91.82(17.06)	94.12(5.86)	65.29(7.14)	16.69(39.67)	17.43
GA	99.38(6.00)	99.38(0.62)	93.35(0.71)	1.22(12.14)	4.87	95.02(20.26)	96.86(3.12)	71.84(0.59)	9.56(46.80)	17.69
IU	95.64(2.26)	95.53(4.47)	87.85(4.79)	7.93(5.43)	4.24	3.69(71.07)	4.03(95.95)	3.85(68.58)	98.8(42.44)	69.51
RL	96.29(2.91)	98.96(1.04)	92.23(0.41)	16.67(3.31)	1.92	68.51(6.25)	98.91(1.07)	69.47(2.96)	85.62(29.26)	9.89
ℓ_1 -sparse	98.71(5.33)	99.88(0.12)	92.92(0.28)	5.29(8.07)	3.45	88.20(13.44)	99.65(0.33)	70.80(1.63)	29.64(26.72)	10.53
SalUn	96.82(3.44)	99.45(0.55)	92.32(0.32)	14.27(0.91)	<u>1.31</u>	82.22(7.46)	98.71(1.27)	67.38(5.05)	66.78(10.42)	<u>6.05</u>
Q-MUL	97.11(3.73)	99.67(0.33)	92.39(0.25)	12.49(0.87)	1.30	75.71(0.95)	97.89(2.09)	67.27(5.16)	52.11(4.25)	3.11
The proportion of forgotten data samples to all samples is 30%										
Retrain	91.61	99.97	91.29	16.11	0	67.67	99.98	67.54	58.47	0
FT	99.19(7.58)	99.95(0.02)	92.94(1.65)	3.10(13.01)	5.57	96.24(28.57)	99.94(0.04)	71.49(3.95)	20.57(37.90)	17.62
GA	99.35(7.74)	99.38(0.59)	93.39(2.10)	1.28(14.83)	6.32	96.73(29.06)	97.30(2.68)	73.35(5.81)	6.19(52.28)	22.46
IU	76.21(15.40)	75.93(24.04)	71.48(19.81)	24.53(8.42)	16.92	2.81(64.86)	3.19(96.79)	3.09(64.45)	96.33(37.86)	65.99
RL	93.96(2.35)	96.03(3.94)	90.34(0.95)	19.07(2.96)	2.55	75.50(7.83)	99.58(0.40)	66.56(0.98)	87.33(28.86)	9.52
ℓ_1 -sparse	99.06(7.45)	99.90(0.07)	93.01(1.72)	4.54(11.57)	5.20	88.59(20.92)	99.79(0.19)	70.72(3.18)	33.16(25.31)	12.40
SalUn	96.17(4.56)	97.78(2.19)	91.57(0.30)	15.50(0.66)	<u>1.93</u>	79.60(11.93)	96.70(3.28)	63.41(4.13)	57.73(0.74)	<u>5.02</u>
Q-MUL	93.57(1.96)	96.77(3.20)	90.24(1.05)	15.59(0.52)	1.68	73.90(6.23)	97.63(2.35)	65.80(1.74)	63.76(5.29)	3.90
The proportion of forgotten data samples to all samples is 50%										
Retrain	90.75	100.0	90.28	18.76	0	65.40	99.99	65.74	68.19	0
FT	99.29(8.54)	99.96(0.04)	93.28(3.00)	2.85(15.91)	6.87	97.10(31.70)	99.98(0.01)	72.82(7.08)	16.86(51.33)	22.71
GA	99.34(8.59)	99.34(0.66)	93.44(3.16)	1.27(17.49)	7.48	96.82(31.42)	97.33(2.66)	73.37(7.63)	5.93(62.26)	25.99
IU	16.12(74.63)	16.02(83.98)	16.22(74.06)	16.36(2.40)	58.77	0.96(64.44)	1.04(98.95)	1.00(64.74)	98.96(30.77)	64.73
RL	95.32(4.57)	97.17(2.83)	90.85(0.57)	21.33(2.57)	2.64	69.20(3.80)	85.60(14.39)	64.39(1.35)	54.58(13.61)	8.29
ℓ_1 -sparse	99.25(8.50)	99.95(0.05)	92.67(2.39)	3.91(14.85)	6.45	96.48(31.08)	99.98(0.01)	72.26(6.52)	25.11(43.08)	20.17
SalUn	96.56(5.81)	98.03(1.97)	91.55(1.27)	18.05(0.71)	<u>2.44</u>	87.75(22.35)	98.99(1.00)	64.69(1.05)	73.12(4.93)	<u>7.33</u>
Q-MUL	91.09 (0.34)	93.94(6.06)	89.12(1.16)	18.68 (0.08)	1.91	67.25(1.85)	89.65(10.34)	61.38(4.36)	56.52(11.67)	7.06

Table 1. Performance of various MU methods for ResNet18 with both activations and weights quantized to 4 bits CIFAR-10 and CIFAR-100. The unlearning scenario is random data forgetting. **Bold** indicates the best performance and underline indicates the runner-up. A performance gap against Retrain is provided in(•).

Method	FA	RA	TA	MIA	AG↓
Retrain	74.76	99.98	72.43	56.36	0
Q-MUL	75.71(0.95)	97.89(2.09)	67.27(5.16)	52.11(4.25)	3.11
w/oSL	73.67(1.09)	96.46(3.52)	65.91(6.52)	51.09(5.27)	4.10
w/oAGR	69.84(4.92)	92.08(7.90)	68.62(3.81)	58.00(1.64)	4.57

Table 2. Impact of Removing Components of Q-MUL. Ablation study is from ResNet18 on CIFAR-100. The unlearning scenario is random data forgetting(10%).

cates that Q-MUL balances the contributions of forgotten and retained data to the model parameter updates. For the MIA metric, Q-MUL is 52.11% (with a gap of 4.25% from Retrain) and SalUn is 66.78% (with a gap of 10.42% from Retrain), showing that the forgetting effect of Q-MUL is closer to the optimal machine unlearning method Retrain. Although the TA metric of Q-MUL has a slightly higher

gap than SalUn, the average gap of the four metrics from Retrain is 3.11%, which is significantly lower than the existing baseline methods. When the proportion of forgotten data samples to all samples is 30% and 50%, Q-MUL can also achieve the smallest average gaps of 3.90% and 7.06%.

In the appendix, we provide experiments on quantized ResNet18 on the larger dataset Tiny-Imagenet and the class-wise forgetting scenario, as well as experiments on quantized MobileNetV2 across different datasets.

5.3. Ablation Studies

The Impact of Different Components. To further demonstrate the effectiveness of different components of Q-MUL, we conduct ablation experiments using two variants of Q-MUL: (a) *w/o* Similar Labels (SL) and (b) *w/o* Adaptive Gradient Reweighting (AGR). As shown in Table 2, using the complete Q-MUL enables the performance of the unlearned model to be closest to that of the retrained model,

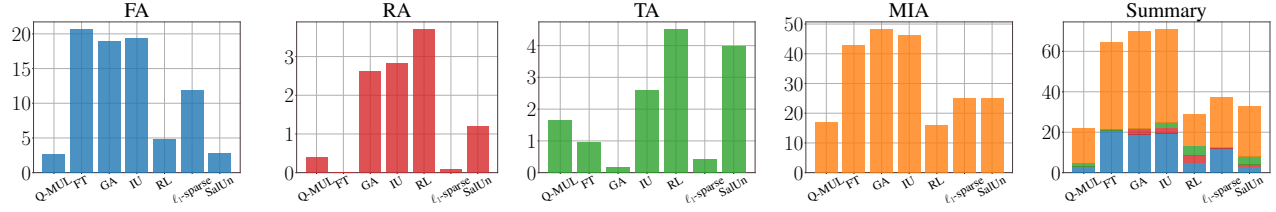


Figure 3. Performance gaps of Full-precision ResNet18 on CIFAR-100. The unlearning scenario is random data forgetting (10%). A shorter bar (smaller gap) indicates performance is closer to that of the Retrained model. The average gap for each method is calculated by dividing the values in the Summary bar chart by 4.

QAT	D_f/D	Bit Width	MU	FA	RA	TA	MIA	AG↓
PACT[6]	10%	4w4a	Retrain	66.64	94.90	65.84	40.78	0
		4w4a	RL	71.87(5.23)	80.28(14.62)	60.56(5.28)	34.18(6.60)	7.93
		4w4a	ℓ_1 -sparse	78.69(12.05)	85.42(9.48)	65.90(0.06)	24.44(16.34)	9.48
		4w4a	SalUn	75.13(8.49)	81.21(13.69)	61.16(4.68)	30.60(10.18)	9.26
		4w4a	Q-MUL	73.51(6.87)	84.73(10.17)	66.10(0.26)	34.09(6.69)	6.00
DSQ [12]	10%	4w4a	Retrain	65.24	99.22	63.62	44.42	0
		4w4a	RL	80.42(15.18)	90.28(8.94)	60.54(3.08)	32.09(12.33)	9.88
		4w4a	ℓ_1 -sparse	89.11(23.87)	97.11(2.11)	65.36(1.74)	19.82(24.60)	13.08
		4w4a	SalUn	82.73(17.49)	87.04(12.18)	58.50(5.12)	21.75(22.67)	14.37
		4w4a	Q-MUL	78.18(12.94)	92.28(6.94)	65.31(1.69)	32.89(11.53)	8.26

Table 3. ResNet18 with both activations and weights quantized to 4 bits on CIFAR-100. We conduct unlearning across different QAT methods. The unlearning scenario is random data forgetting(10%).

with an average gap of only 3.11%. When SL is removed, the gap in the RA metric increases from 2.09% to 3.52%, indicating a decline in the discriminative ability of the unlearned model on the retained data. This is attributed to the introduction of noise inconsistent with the original data during the data processing phase, which leads to excessive model adjustments and the forgetting of information that should not have been forgotten. When AGR is removed, the forgotten model’s FA and RA are 69.84% and 92.08%, respectively, with gaps of 4.92% and 7.90% compared to the retrained model’s FA and RA (which increased by 3.97% and 5.81% when using AGR, respectively). This indicates that AGR is crucial for balancing the contributions of forgotten and retained data to the model parameter updates.

Q-MUL with Full-precision Models. To more comprehensively evaluate Q-MUL, we compare the effects of different unlearning methods on the full-precision model, as shown in Figure 3. The issues of noise introduction and gradient imbalance also exist in the full-precision model, although they are not as severe as in the quantized model. However, Q-MUL can still address these two issues. Specifically, the average gap of Q-MUL can reach 5.46%, which is significantly lower than that of existing methods. This demonstrates that Q-MUL can achieve superior unlearning performance even in full-precision models.

Q-MUL with Different QAT Methods. In addition to

LSQ+, we also validate the performance of different MU methods under other QAT methods. As shown in Table 3, Q-MUL can achieve SOTA performance in other QAT methods as well. Specifically, for PACT, the gaps between Q-MUL and Retrain in FA and TA are 6.87% and 10.17%, respectively, with an average gap of 6.00%. For DSQ, the gaps between Q-MUL and Retrain in FA and TA are 12.94% and 6.94%, respectively, with an average gap of 8.26%. Q-MUL effectively reduces the RA gap while balancing the FA gap and RA gap as much as possible.

6. Conclusions

This paper is the first to directly apply machine unlearning to quantized models. We analyze the challenges of machine unlearning in quantized models: noise introduced during the data processing phase and the large gradient difference of the loss function between forgotten and retained data. To address these challenges, we propose Q-MUL. This method reduces noise by replacing the labels of forgotten data with similar labels during the data processing phase. During training, it adaptively adjusts the weights of the loss from forgotten and retained data based on the gradient to balance their contributions to model parameter updates. Experimental results show that Q-MUL is more suitable for quantized models than existing methods.

References

- [1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3
- [2] Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 696–697, 2020. 2, 6
- [3] Lucas Bourtole, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021. 1, 2
- [4] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13169–13178, 2020. 2
- [5] Eli Chien, Chao Pan, and Olga Milenkovic. Certified graph unlearning. *arXiv preprint arXiv:2206.09140*, 2022. 1
- [6] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018. 2, 8
- [7] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. In *International Conference on Learning Representations*, 2020. 2
- [8] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023. 1, 2, 3, 5, 6
- [9] Jun Fang, Ali Shafiee, Hamzah Abdel-Aziz, David Thorsley, Georgios Georgiadis, and Joseph H Hassoun. Post-training piecewise linear quantization for deep neural networks. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 69–86. Springer, 2020. 1
- [10] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019. 2
- [11] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020. 1, 2, 3, 5, 6
- [12] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4852–4861, 2019. 2, 8
- [13] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11516–11524, 2021. 6
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 6
- [15] Chris Jay Hoofnagle, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98, 2019. 1
- [16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 6
- [17] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *International Conference on Artificial Intelligence and Statistics*, pages 2008–2016. PMLR, 2021. 6
- [18] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017. 6
- [19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6
- [20] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6
- [21] Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, PRANAY SHARMA, Sijia Liu, et al. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36, 2024. 6
- [22] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020. 1, 2
- [23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bisacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, page 4. Granada, 2011. 6
- [24] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*, 2022. 3
- [25] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021. 2
- [26] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017. 6

- [27] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE, 2022. [6](#)
- [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. [1](#)
- [29] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021. [6](#)
- [30] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhong Cao, Chuangrun Liang, and Minghui Tan. Generative low-bitwidth data free quantization. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 1–17. Springer, 2020. [2](#)
- [31] Xiangguo Zhang, Haotong Qin, Yifu Ding, Ruihao Gong, Qinghua Yan, Renshuai Tao, Yuhang Li, Fengwei Yu, and Xianglong Liu. Diversifying sample generation for accurate data-free quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15658–15667, 2021. [2](#)
- [32] Yiren Zhou, Seyed-Mohsen Moosavi-Dezfooli, Ngai-Man Cheung, and Pascal Frossard. Adaptive quantization for deep neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. [1](#)