# Ferrari: Federated Feature Unlearning via Optimizing Feature Sensitivity

Hanlin Gu[2*], Win Kent Ong[1*], Chee Seng Chan[1†], and Lixin Fan[2]

[1]CISiP, Universiti Malaya, Malaysia
[2]AI Lab, Webank, PR China

## Abstract

The advent of Federated Learning (FL) highlights the practical necessity for the *'right to be forgotten'* for all clients, allowing them to request data deletion from the machine learning model's service provider. This necessity has spurred a growing demand for Federated Unlearning (FU). Feature unlearning has gained considerable attention due to its applications in unlearning sensitive, backdoor, and biased features. Existing methods employ the influence function to achieve feature unlearning, which is impractical for FL as it necessitates the participation of other clients, *if not all*, in the unlearning process. Furthermore, current research lacks an evaluation of the effectiveness of feature unlearning. To address these limitations, we define feature sensitivity in evaluating feature unlearning according to Lipschitz continuity. This metric characterizes the model output's rate of change or sensitivity to perturbations in the input feature. We then propose an effective federated feature unlearning framework called Ferrari, which minimizes feature sensitivity. Extensive experimental results and theoretical analysis demonstrate the effectiveness of Ferrari across various feature unlearning scenarios, including sensitive, backdoor, and biased features. The code is publicly available at https://github.com/OngWinKent/Federated-Feature-Unlearning

## 1 Introduction

Federated Learning (FL) [1–3] allows for model training across decentralized devices or servers holding local private data samples, without the need to exchange them directly. An essential requirement within FL is the participants *"right to be forgotten"*, as explicitly outlined in regulations such as the European Union General Data Protection Regulation (GDPR)[3] and the California Consumer Privacy Act (CCPA)[4] [4]. To address this requirement, Federated Unlearning (FU) has been introduced, enabling clients to selectively remove the influence of specific subsets of their data from a trained FL model while preserving the model's accuracy on the remaining data [5].

Different from unlearning at the *client, class, or sample* level [6–8] in FL, the feature unlearning [9] holds significant applications across various scenarios. Firstly, in contexts where sentences contain sensitive information such as names and addresses [9, 10], it becomes crucial to remove these sensitive components to prevent potential exposure through model inversion attacks [11–14]. Secondly, when datasets contain backdoor triggers that can compromise model integrity [15–18], it is imperative to eliminate these patterns. Thirdly, unlearning biased features becomes essential in scenarios where data imbalances significantly impact model accuracy due to bias [19–22]. However,

[*]equal contribution; authors are listed alphabetically by first name.
[†]corresponding author (*cs.chan@um.edu.my*).
[3]`https://gdpr-info.eu/art-17-gdpr/`
[4]`https://oag.ca.gov/privacy/ccpa`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

existing works of FU focus on client, class, or sample unlearning [6–8] but do not address feature unlearning, limiting their ability to unlearn specific features across multiple data points.

There are two challenges in feature unlearning in FL. Firstly, evaluating the unlearning effectiveness for feature unlearning is difficult. Typically, unlearning effectiveness is assessed by comparing the unlearned model with a retrained model without the feature. However, building data without the feature is challenging; for example, training the data with noise or a black block on the feature region may cause severe degradation in model accuracy (see Sec. 3.2). Secondly, previous work on feature unlearning within centralized machine learning settings [9, 10] is not practical for FL due to its requirement for access to all datasets, necessitating the participation of all clients.

To address the aforementioned limitations, we first define the feature sensitivity in Sec. 4.1 to evaluate the feature unlearning inspired by the Lipschitz continuity, which characterizes the rate of change or sensitivity of the model output to perturbations in the input feature. Then we propose a simple but effective federated feature unlearning method, called Ferrari (**F**ed**er**ated Featu**r**e Unle**arni**ng), by minimizing the feature sensitivity in Sec. 4.2. Our Ferrari framework offers three key advantages: Firstly, Ferrari requires only local datasets from the unlearned clients for feature unlearning. Secondly, Ferrari demonstrates high practicality and efficiency, which support various feature unlearning scenarios, including sensitive, backdoor, and biased features and only consumes a few epochs of optimization. Thirdly, theoretical analysis in Sec. 4.3 elucidates that our proposed Ferrari achieves lower model utility loss compared to the exact feature unlearning.

The key contributions of this work are summarized as follows:

- We identify two key challenges for feature unlearning in FL. The first is how to successfully unlearn features without requiring the participation of other clients, as discussed in Sec. 3.2. The second is how to design an effective evaluation method in federated feature unlearning.
- We define the feature sensitivity and introduce this metric in federated feature unlearning in Sec. 4. By minimizing feature sensitivity, we propose an effective federated feature unlearning method, named Ferrari, which enables clients to selectively unlearn specific features from the trained global model without requiring the participation of other clients.
- We provide a theoretical proof in Theorem 1, which dictates that *Ferrari achieves better model performances than exact feature unlearning*. This analytical result is also echoed in the empirical evidence, highlighting Ferrari's effectiveness across various settings, including the unlearning of sensitive, backdoor, and biased features.

## 2 Related Work

**Machine Unlearning** Machine Unlearning (MU), introduced by Cao et al. [23], involves selectively removing specific training data from a trained model without retraining from scratch[24, 25]. It categorizes into exact unlearning [26, 27], aiming to completely remove data influence with techniques like SISA [28] and ARCANE [29], though with computational costs, and approximate unlearning [30, 31], which reduces data impact through techniques like data manipulation (fine-tuning with mislabeled data [32–36] or introducing noise [37–39]), knowledge distillation [40–43] (training a student model), gradient ascent [44–47] (maximizing loss associated with forgotten data), and weight scrubbing [48–53] (discarding heavily influenced weights).

**Federated Unlearning** In FL, traditional centralized MU methods are unsuitable due to inherent differences like incremental learning and limited dataset access [54]. Research on Federated Unlearning (FU) mainly focuses on client, class, and sample unlearning [6–8]. Client unlearning, pioneered by Liu et al. [55] introducing FedEraser [55], includes approaches like FRU [56], FedRecover [57], VeriFI [58], HDUS [59], KNOT [60], FedRecovery [61], Knowledge Distillation [54], and Gradient Ascent [62–64], aiming to remove specific clients or recover poisoned global models. Class unlearning, introduced by Wang et al. [65], involves frameworks like discriminative pruning and Momentum Degradation [66] (MoDE) to remove entire data classes. Sample unlearning, initiated by Liu et al. [67], targets individual sample removal within FL settings, with advancements like the QuickDrop [68] framework and FedFilter [69] enhancing efficiency and effectiveness. Recent works, such as $FedMe^2$ by Xia et al. [70], optimize both unlearning facilitation and privacy guarantees.

Existing literature on FU primarily focuses on client, class, or sample unlearning [6–8]. However, a significant gap arises when a client seeks to remove only sensitive features while remaining engaged in FL. Unfortunately, current FU approaches do not address this specific scenario, as they do not explore feature unlearning within FL settings. In contrast to prior works focusing on feature unlearning in centralized settings of MU, such as classification models [9, 10], generative models [71–74], and large language models [75–77], this study uniquely addresses feature unlearning of classification model within the FL paradigm. This distinction arises because traditional feature unlearning methods in centralized settings of MU are impractical for FL scenarios, where participation from all clients is often infeasible. In such cases, the process fails if even a single client opts out of the operation.

Therefore, to fill this critical gap, we proposed a novel federated feature unlearning framework, namely Ferrari based on the concept of Lipschitz continuity [78–80]. Our proposed Ferrari requires exclusively from the target client's dataset while still preserving the model's original performance. Lipschitz continuity, a fundamental mathematical concept that measures a function's sensitivity to changes in its input variables [81–83], is central to our feature unlearning approach. For a detailed exposition of our proposed federated feature unlearning framework utilizing Lipschitz continuity, please refer to Sec. 4. To the best of our knowledge, this is the **first work** in feature unlearning within FL settings that does not necessitate participation from all other clients, showcasing the potential to enhance privacy, practicality and efficiency.

# 3 Challenges on Feature Unlearning in FL

## 3.1 Federated Feature Unlearning

Consider a federated system comprising $K$ clients and one server, collaboratively learning a global model $f_\theta$ as:

$$\min_{\theta} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{\ell(f_\theta(x_{k,i}), y_{k,i})}{n_1 + \cdots + n_K}, \tag{1}$$

where $\ell$ is the loss, *e.g.,* the cross-entropy loss, $\mathcal{D}_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}$ is the dataset with size $n_k$ owned by client $k$. One client (*i.e.,* referred to as the unlearn client $C_u$) requests the removal of a feature $\mathcal{F}$ from the global model $\theta$ such that $\theta$ does not retain any information about $\mathcal{F}$. Specifically, we assume that the data $x \in \mathbb{R}^d$ and denote the j-th feature of $x$ by $x[j]$. The partial element of the data $x$ corresponding the feature $\mathcal{F}$ is defined as $x[\mathcal{F}]$, *i.e.,*:

$$x[\mathcal{F}] = \{x[j], j \in \mathcal{F}\} \tag{2}$$

Therefore, the unlearn client $C_u$ aims to remove $\{x_{i,u}[\mathcal{F}]\}_{i=1}^{n_u}$, called unlearned data $\mathcal{D}_u$. Denote $\mathcal{D}_r = \mathcal{D} - \mathcal{D}_u$ to be the remaining data.

## 3.2 Challenges for Feature Unlearning in FL

Unlike sample or class unlearning [6–8], evaluating the unlearning effectiveness for feature unlearning is difficult. Typically, unlearning effectiveness is assessed by comparing the unlearned model with a retrained model trained on remaining data $\mathcal{D}_r$. However, building $\mathcal{D}_r$ for the feature unlearning takes much work. For example, suppose we want to remove the mouth from a face image. In that case, one possible solution is to replace the mouth region with Gaussian noise or black block, as illustrated in Fig. 1. However, this added Gaussian noise or black block can adversely affect model training and degrade performance, *e.g.,* the degradation of model accuracy is beyond 27%.

$x$ Acc 95.86% | $\overline{x}_G$ 75.51% | $\overline{x}_B$ 68.37%

Figure 1: Sample data $x$ with Gaussian noise ($\overline{x}_G$) and black pixels ($\overline{x}_B$) perturbations, illustrating feature removal and performance comparison.

Another challenge is implementing feature unlearning for $C_u$ without the help of other clients. Previous work on feature unlearning [9, 10] typically requires access to the remaining data, necessitating the participation of other clients in the FL process. This requirement is impractical in the FL context, as other clients may be unwilling or unable to share data or computational resources. Therefore, finding a method to effectively unlearn features without relying on other clients is crucial to maintain the model accuracy and practicality in the FL settings.

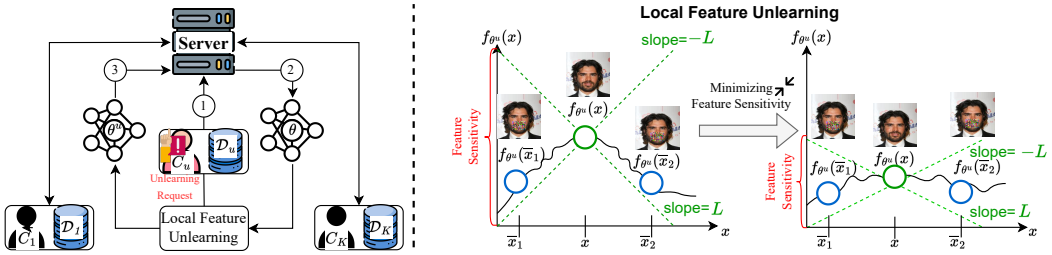



Figure 2: Overview of our proposed Ferrari framework: Initiated by the feature unlearning request from the unlearn client $C_u$, the server initializes the trained global model $\theta$ to $C_u$ for local feature unlearning. Upon completion, $C_u$ uploads the unlearned model $\theta^u$ to the server. Local feature unlearning minimizes the Lipschitz constant $L$ between the original input and its perturbed feature subset, reducing feature sensitivity yet preserving the overall model performance.

# 4 The Proposed Method

In this section, we introduce feature sensitivity (see Def. 1) in Sec. 4.1 to evaluate the effectiveness of feature unlearning. We then propose Ferrari based on this concept in Sec. 4.2). Finally, we demonstrate that Ferrari achieves a lower utility loss compared to exact feature unlearning in Sec. 4.3).

## 4.1 Feature Sensitivity

Inspired by Lipschitz Continuity [79, 80, 82], which provides an approximate method for removing information from images by perturbing the input data and observing the effect on the output, we introduce the concept of **feature sensitivity** $s$ as Def. 1. This metric measures the memorization of a model $f_\theta$ for the feature $\mathcal{F}$ by considering the local changes in the given input rather than the global change as defined in the traditional Lipschitz continuity.

**Algorithm 1** Federated Feature Unlearning

**Input:** Unlearn client $C_u$, Local dataset $\mathcal{D}_u$ with data size $n_u$, Unlearn feature $\{\mathcal{F}_i\}_{i=1}^N$, Global model parameters $\theta$, Gaussian noise $\sigma$, Learning rate $\eta$, Sample number $N$
**Output:** Unlearned model parameters $\theta^u$

▷ *The unlearn client $C_u$ performs:*
**for** $(x, \mathcal{F}_i)$ in $(\mathcal{D}_u, \{\mathcal{F}_i\}_{i=1}^N)$ **do**
  $\theta^u = \theta$
  **for** $i = 1$ **to** $N$ **do**
    Sample $\delta_{\mathcal{F}_i}$ according to Eq. (4)
    Compute $L_i = \frac{\|f_{\theta^u}(x) - f_{\theta^u}(x+\delta_{\mathcal{F},i})\|_2}{\|\delta_{\mathcal{F},i}\|_2}$
  **end for**
  $L = \frac{1}{N}\sum_{i=1}^N L_i$
  $\theta^u \leftarrow \theta^u - \eta \cdot \nabla_{\theta^u}(L)$
**end for**
Upload $\theta^u$ to the server
▷ *The server performs:*
Replace the global model $\theta$ with the $\theta^u$
**return** $\theta^u$

**Definition 1.** *The feature sensitivity $s$ of the model $f$ with respect to the feature $\mathcal{F}$ on the data $(x, y)$ is defined as:*

$$s = \mathbb{E}_{\delta_\mathcal{F}} \frac{\|f(x) - f(x + \delta_\mathcal{F})\|_2}{\|\delta_\mathcal{F}\|_2}, \qquad (3)$$

*where $\delta_\mathcal{F}$ denote the perturbation on feature $\mathcal{F}$.*

Def. 1 characterizes the rate of change or sensitivity of the model output to perturbations in the input data. A small feature sensitivity $s$ represents the model $f$ doesn't memorize the feature $\mathcal{F}$. This definition does not require building the remaining data, as it considers the expectation over the perturbation $\delta_\mathcal{F}$. Specifically, it represents the average output change rate over any magnitude of the perturbation. Furthermore, we will provide the relationship between Def. 1 and exact feature unlearning in Sec. 4.3.

**Remark 1.** *The perturbation $\delta_\mathcal{F}$ can be chosen from various distributions, such as the Gaussian distribution, the uniform distribution, and so on.*

## 4.2 Ferrari

As discussed the feature sensitivity $s$ in Sec. 4.1, the core idea of the proposed method Ferrari is to achieve the feature unlearning by minimizing the feature sensitivity. More specifically, it controls the change in the model's output relative to changes in the input within the feature region, *i.e.,* the slope, to prevent the model from memorizing the feature as illustrated in Fig. 2.

One unlearning client $C_u$ requests to unlearning the feature $\mathcal{F}$. The proposed Ferrari aims to unlearn the global model $\theta$ to $\theta^u$. The proposed method can be divided into three steps (see details in Alg. 1). In order to compute the feature sensitivity, the perturbation $\delta_{\mathcal{F}}$ in terms of the feature $\mathcal{F}$ is **firstly** computed as the following (take the Gaussian distribution as an example):

$$\delta_{\mathcal{F}}[j] = \begin{cases} \sim N(0, \sigma^2) & j \in \mathcal{F} \\ 0 & \text{Otherwise} \end{cases} \tag{4}$$

**Secondly,** we leverage a finite sample Monte Carlo approximation to the maximization as Def. 1 as:

$$\mathbb{E}_{\delta_{\mathcal{F}}} \frac{\|f_\theta(x) - f_\theta(x + \delta_{\mathcal{F}})\|_2}{\|\delta_{\mathcal{F}}\|_2} \sim \frac{1}{N} \sum_{i=1}^{N} \frac{\|f_\theta(x) - f_\theta(x + \delta_{\mathcal{F},i})\|_2}{\|\delta_{\mathcal{F},i}\|_2}, \tag{5}$$

where $\delta_{\mathcal{F},i}$ is $i_{th}$ sampling as Eq. (4).

**Finally**, for the unlearning client $C_u$ who aims to remove the feature $\mathcal{F}$ from his/her data $\mathcal{D}_u$, the unlearned model $\theta^u$ is obtained as the following:

$$\theta^u = \arg\min_{\theta} \mathbb{E}_{(x,y)\in\mathcal{D}_u} \frac{1}{N} \sum_{i=1}^{N} \frac{\|f_\theta(x) - f_\theta(x + \delta_{\mathcal{F},i})\|_2}{\|\delta_{\mathcal{F},i}\|_2}, \tag{6}$$

where Eq. (6) is computed over the dataset $\mathcal{D}_u$. Noted that the proposed Ferrari based on Def. 1 doesn't need the participation of other clients.

**Remark 2.** *When the unlearning happens during the federated training, the unlearning clients would also optimize the training loss and feature sensitivity simultaneously, i.e.,,* $\mathbb{E}_{(x,y)\in\mathcal{D}}\big(\ell(f_\theta(x), y) + \lambda \mathbb{E}_{\delta_{\mathcal{F}}} \frac{\|f_\theta(x) - f_\theta(x+\delta_{\mathcal{F}})\|_2}{\|\delta_{\mathcal{F}}\|_2}\big)$*, where* $\lambda$ *is a coefficient.*

### 4.3 Theoretical Analysis of the Utility loss for Ferrari

As illustrated in Sec. 3.2, retraining the model without the feature may affect the model accuracy seriously. Suppose the feature is successfully removed when the norm of perturbation is larger than $C$. We firstly define the utility loss $\ell_1$ with unlearning feature directly, *i.e.,*, **the exact feature unlearning**:

$$\ell_1 = \min_{\|\delta_{\mathcal{F}}\| \geq C} \mathbb{E}_{(x,y)\in\mathcal{D}} \min_{\theta} \ell\big(f_\theta(x + \delta_{\mathcal{F}}), y\big) \tag{7}$$

And we define the maximum utility loss with the norm perturbation lower than $C$ as:

$$\ell_2 = \max_{\|\delta_{\mathcal{F}}\| \leq C} \mathbb{E}_{(x,y)\in\mathcal{D}} \min_{\theta} \ell\big(f_\theta(x + \delta_{\mathcal{F}}), y\big) \tag{8}$$

**Assumption 1.** *Assume* $\ell_2 \leq \ell_1$

Assumption 1 elucidates that the utility loss associated with a perturbation norm lower than $C$ is smaller than the utility loss when the perturbation norm is greater than $C$. This assumption is logical, as larger perturbations would naturally lead to a greater utility loss.

**Assumption 2.** *Suppose the federated model achieves zero training loss.*

We have the following theorem to elucidate the relation between feature sensitivity removing via Alg. 1 and exact unlearning (see proof in Appendix A.1, including the extension for the non-zero training loss assumption).

**Theorem 1.** *If Assumptions 1 and 2 hold, the utility loss of unlearned model obtained using Alg. 1 is lower than the utility loss with exact feature unlearning, i.e.,,*

$$\ell_u \leq \ell_1, \tag{9}$$

*where* $\ell_u = \mathbb{E}_{(x,y)\in\mathcal{D}} \ell(f_{\theta^u}(x), y)$

Theorem 1 showcases that the proposed method Ferrari, results in a utility loss ($\ell_u$) that is lower than the utility loss incurred when the feature is removed, and the model is retrained, *i.e.,* the process of exact feature unlearning.

**Remark 3.** *To further evaluate the effectiveness of feature unlearning based on feature sensitivity, we employ model inversion attacks [11, 12] to determine if the feature can be reconstructed and employ attention maps to assess if the model still focuses on the unlearned feature, as described in Sec. 5.3.1.*

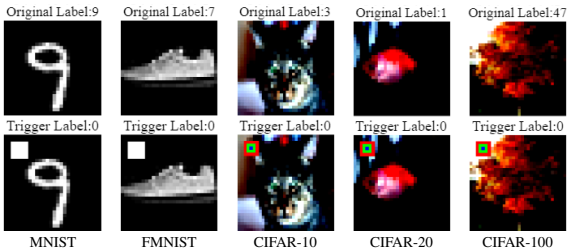



Figure 3: Pixel-pattern backdoor feature.

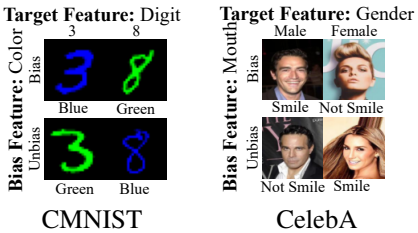



Figure 4: Biased datasets distribution.

## 5 Experimental Results

This section presents the empirical analysis of the proposed Ferrari framework in terms of effectiveness, utility, and time efficiency in sensitive, backdoor and biased feature unlearning scenarios.

### 5.1 Experimental Setup

**Unlearning Scenarios** *Sensitive Feature Unlearning*: We simulate the removal of sensitive features from the $\mathcal{D}_u$ to fulfill the request of $C_u$ due to privacy concern. Specifically, we remove 'mouth' from CelebA [84], 'marital status' from Adult [85], and 'pregnancies number' from Diabetes [86]. Therefore, our proposed Ferrari aims to remove the influence of these requested features.

*Backdoor Feature Unlearning*: We simulate a pixel-pattern backdoor attack by $C_u$ based on BadNets [18] within a FL framework [15–17]. $C_u$ injects a pixel-pattern backdoor feature and trigger label into its $\mathcal{D}_u$ during training, as shown in Fig. 3. Consequently, our proposed Ferrari aims to remove the influence of these backdoor features and restore the model's original performance.

*Biased Feature Unlearning*: We simulate the bias dataset $\mathcal{D}_u$ of the $C_u$ and the unbias dataset $\mathcal{D}_r$ with a bias ratio of 0.8, as shown in Fig. 4. This results in a global model biased towards the biased dataset [87, 88] due to unintended feature memorization [22]. In CMNIST [89], the model focuses on color patterns instead of digits, and in CelebA [84], it learns mouth features instead of facial features for gender classification. Therefore, our proposed Ferrari aims to mitigate these bias-inducing features and restore model performance.

**Hyperparameters & Datasets & Model** We simulate HFL with $K = 10$ clients under an IID setting, each holding $10\%$ of the datasets, except for the biased feature unlearning experiment with a bias ratio of 0.8. For federated feature unlearning experiments, we set hyperparameters: learning rate $\eta = 0.0001$, sample size $N = 20$, and random Gaussian noise with standard deviation ranging from $0.05 \leq \sigma \leq 1.0$ (see Sec. 5.5) across iterations of $N$. Experiments are repeated over five random trials, and results are reported as mean and standard deviation. We employ ResNet18 [90] on image datasets: MNIST [89], Colored-MNIST (CMNIST) [89], Fashion-MNIST [91], CIFAR-10, CIFAR-20, CIFAR-100 [92] and ImageNet [93]. For tabular datasets, such as Adult Census Income (Adult) [85] and Diabetes [86], we used a fully-connected neural network linear model. Additionally, we utilize the transformer-based BERT model [94] for the text dataset, specifically the IMDB movie reviews dataset [95]. We conduct experiments on a single NVIDIA A100 GPU. Further details are in Appendix A.2.

**Evaluation Metrics** We assess effectiveness by measuring feature sensitivity (see Section 4.1) and conducting a model inversion attack (MIA) [11–14] to determine the attack success rate (ASR). The goal is to achieve low feature sensitivity and ASR, indicating successful unlearning sensitive features. Backdoor and biased feature unlearning are evaluated by comparing accuracy on the retain dataset $\mathcal{D}_r$ ($Acc_r$) and the unlearn client dataset $\mathcal{D}_u$($Acc_u$). Low $Acc_u$ indicates high effectiveness for backdoor unlearning, while similar accuracy ($Acc_r \approx Acc_u$) reflects fairness and effectiveness in biased feature unlearning. Qualitatively, effectiveness is assessed using MIA-reconstructed images (sensitive) and GradCAM [96] attention maps (backdoor and biased). The utility is measured by test dataset $\mathcal{D}_t$ accuracy ($Acc_t$), with higher values indicating stronger utility. Time efficiency is evaluated by comparing the runtime of each baseline.

| Scenarios | Datasets | Unlearn Feature | Accuracy(%) | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Baseline | Retrain | Fine-tune | FedCDP[65] | FedRecovery[61] | Ferrari (Ours) |
| Sensitive | **CelebA** | Mouth | 94.87 ±1.38 | 79.46 ±2.32 | 62.79 ±1.62 | 34.03 ±4.20 | 29.78 ±6.69 | **92.26 ±1.73** |
| | **Adult** | Marriage | 82.45 ±2.59 | 65.27 ±0.58 | 61.02 ±1.05 | 30.19 ±1.62 | 27.89 ±3.71 | **81.02 ±0.58** |
| | **Diabetes** | Pregnancies | 82.11 ±0.49 | 64.19 ±0.72 | 59.57 ±0.68 | 36.71 ±4.56 | 17.56 ±2.32 | **79.53 ±0.79** |
| | **IMDB** | Names | 91.39 ±1.57 | 83.27 ±2.05 | 72.15 ±1.92 | 48.36 ±2.79 | 37.93 ±2.84 | **89.15 ±1.32** |
| Backdoor | **MNIST** | Backdoor Pixel Pattern | 94.75 ±4.88 | 96.23 ±0.16 | **96.85 ±0.91** | 65.31 ±4.39 | 40.52 ±7.38 | 95.83 ±1.14 |
| | **FMNIST** | | 90.68 ±2.19 | 92.98 ±0.75 | **93.52 ±1.63** | 67.62 ±0.81 | 42.24 ±4.45 | 92.61 ±1.57 |
| | **CIFAR-10** | | 87.55 ±3.71 | 90.92 ±1.83 | **91.23 ±0.44** | 53.98 ±2.17 | 27.16 ±9.68 | 89.52 ±2.18 |
| | **CIFAR-20** | | 74.47 ±2.38 | 81.61 ±1.75 | **82.52 ±0.69** | 54.76 ±0.98 | 23.02 ±3.11 | 78.34 ±2.35 |
| | **CIFAR-100** | | 54.13 ±7.62 | 73.12 ±1.54 | **73.59 ±1.66** | 34.30 ±0.42 | 15.21 ±5.83 | 69.30 ±2.27 |
| | **ImageNet** | | 52.86 ±4.14 | 67.18 ±2.07 | **67.52 ±1.69** | 31.17 ±3.96 | 12.75 ±5.27 | 65.36 ±1.84 |
| Biased | **CMNIST** | Color | 81.72 ±3.41 | 98.49 ±1.46 | 82.54 ±0.78 | 27.56 ±1.71 | 25.05 ±5.09 | **83.85 ±1.63** |
| | **CelebA** | Mouth | 87.35 ±4.07 | 95.87 ±1.52 | 88.93 ±2.65 | 16.98 ±0.23 | 20.19 ±7.21 | **94.62 ±2.49** |

Table 1: The accuracy of $\mathcal{D}_t$ for each unlearning method across different unlearning scenarios.

| Scenario | Datasets | Unlearn Feature | Feature Sensitivity | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Baseline | Retrain | Fine-tune | FedCDP [65] | FedRecovery [61] | Ferrari (Ours) |
| Sensitive | **CelebA** | Mouth | $0.96 \pm 1.41 \times 10^{-2}$ | $0.07 \pm 8.06 \times 10^{-4}$ | $0.79 \pm 2.05 \times 10^{-2}$ | $0.93 \pm 2.87 \times 10^{-2}$ | $0.91 \pm 3.41 \times 10^{-2}$ | $\mathbf{0.09 \pm 3.04} \times 10^{-4}$ |
| | **Adult** | Marriage | $1.31 \pm 1.53 \times 10^{-2}$ | $0.02 \pm 6.47 \times 10^{-4}$ | $0.94 \pm 6.81 \times 10^{-2}$ | $1.07 \pm 7.43 \times 10^{-2}$ | $1.14 \pm 2.57 \times 10^{-2}$ | $\mathbf{0.05 \pm 1.72} \times 10^{-4}$ |
| | **Diabetes** | Pregnancies | $1.52 \pm 0.91 \times 10^{-2}$ | $0.05 \pm 5.07 \times 10^{-4}$ | $0.96 \pm 1.28 \times 10^{-2}$ | $1.23 \pm 3.82 \times 10^{-2}$ | $0.83 \pm 5.08 \times 10^{-2}$ | $\mathbf{0.07 \pm 1.07} \times 10^{-4}$ |
| | **IMDB** | Names | $0.85 \pm 1.07 \times 10^{-2}$ | $0.07 \pm 5.38 \times 10^{-4}$ | $0.74 \pm 3.81 \times 10^{-2}$ | $0.81 \pm 3.27 \times 10^{-2}$ | $0.78 \pm 2.41 \times 10^{-2}$ | $\mathbf{0.08 \pm 1.32} \times 10^{-4}$ |

Table 2: Feature sensitivity for each unlearning method across sensitive feature unlearning scenario.

**Baselines** We compare our proposed Ferrari against the models of Baseline, Retrain, Fine-tune, FedCDP [65] and FedRecovery [61]. Additional details are provided in Appendix A.2.

### 5.2 Utility Guarantee

To evaluate the utility of Ferrari, we measure $Acc_t$ on $\mathcal{D}_t$, where a higher $Acc_t$ indicates greater utility (Tab. 1). Although the Fine-tune method shows high $Acc_t$ in the backdoor feature unlearning scenario with a clean dataset, its unlearning effectiveness is very low (see Sec. 5.3.2). This problem worsens with FedCDP [65] and FedRecovery [61], which suffer significant $Acc_t$ declines, reducing model utility and making them unsuitable for feature unlearning. In contrast, Ferrari achieves the highest model utility in sensitive and biased feature unlearning scenarios, with the highest $Acc_t$ among baselines, minimal deterioration, and the greatest unlearning effectiveness across all scenarios.

### 5.3 Effectiveness Guarantee

In this subsection, we analyze the unlearning effectiveness of Ferrari against baselines in sensitive, backdoor, and biased feature unlearning scenarios.

#### 5.3.1 Sensitive Feature Unlearning

To evaluate Ferrari's effectiveness in unlearning sensitive features, we measured feature sensitivity (see Sec. 4.1) and conducted a model inversion attack (MIA) [11–14].

**Feature Sensitivity** Tab. 2 shows the sensitivity of the unlearn feature. The baseline model had high sensitivity to this feature. Similar results were observed for the Fine-tune, FedCDP [65], and FedRecovery models [61], with sensitivities greater than 0.8, indicating ineffective unlearning. In contrast, our proposed Ferrari model exhibits low sensitivity, similar to the Retrain model, indicating successful unlearning of the sensitive feature.

**ASR of MIA** Tab. 3 shows the ASR results. The Baseline model achieved an ASR exceeding 80%, indicating substantial exposure of sensitive features. Similar observations were made for the Fine-tune, FedCDP [65], and FedRecovery [61] models, with ASR surpassing 70% exhibiting ineffective feature unlearning. Conversely, Ferrari achieved low ASR, suggesting successful feature unlearning with minimal unlearned feature exposure after using Ferrari via MIA.

**MIA Reconstruction** Fig. 5 shows MIA-reconstructed images. The Baseline model achieved complete reconstruction, whereas both Retrain and Ferrari models failed to reconstruct the mouth feature accurately. This underscores Ferrari's effectiveness in unlearning and preserving privacy by preventing precise reconstruction of unlearned features via MIA.

| Scenario | Datasets | Unlearn Feature | Attack Success Rate(ASR) (%) | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Baseline | Retrain | Fine-tune | FedCDP [65] | FedRecovery [61] | Ferrari (Ours) |
| Sensitive | **CelebA** | Mouth | 84.36 ±3.22 | 47.52 ±1.04 | 77.43 ±10.98 | 75.36 ±9.31 | 71.52 ±6.07 | **51.28 ±2.41** |
| | **Adult** | Marriage | 87.54 ±13.89 | 49.28 ±2.13 | 83.45 ±8.44 | 72.83 ±5.18 | 80.39 ±10.68 | **49.58 ±1.38** |
| | **Diabetes** | Pregnancies | 92.31 ±7.55 | 38.89 ±2.52 | 88.46 ±5.01 | 81.91 ±8.17 | 78.27 ±2.47 | **42.61 ±1.81** |
| | **IMDB** | Names | 90.28 ±2.49 | 40.29 ±1.59 | 86.74 ±3.81 | 83.67 ±4.59 | 80.95 ±3.51 | **43.75 ±1.86** |

Table 3: The ASR of MIA for each unlearning method across sensitive feature unlearning scenario.

Figure 5: MIA reconstruction on CelebA (unlearned mouth)

### 5.3.2 Backdoor Feature Unlearning

**Accuracy** $\mathcal{D}_r$ and $\mathcal{D}_u$ represent the clean and backdoor datasets, respectively. Successful unlearning is shown by low $Acc_u$ and high $Acc_r$, indicating effective unlearning and preserved model utility. As shown in Tab. 4, the Fine-tune method has higher $Acc_r$ and utility than the Retrain method but lower unlearning effectiveness due to high $Acc_u$. FedCDP [65] and FedRecovery [61] show low utility and unlearning effectiveness with low $Acc_r$ and $Acc_u$, rendering them unsuitable for backdoor feature unlearning. In contrast, Ferrari demonstrates the highest utility and unlearning effectiveness.

**Attention Map** Fig. 6a illustrates attention maps analyzing backdoor feature unlearning. Initially, the Baseline model focuses on the $5 \times 5$ square at the top-left corner, indicating a significant influence on output prediction by the pixel-pattern backdoor feature. In contrast, Ferrari unlearned models shift the attention towards recognizable objects like digits and cars, similar to the Retrain model. This shift suggests a reduced sensitivity to the backdoor feature, indicating a successful unlearning. See Appendix A.3.1 for supplementary results.

### 5.3.3 Biased Feature Unlearning

**Accuracy** $\mathcal{D}_r$ and $\mathcal{D}_u$ represent the unbias and bias datasets, respectively. Successful unlearning results in similar accuracies across both datasets ($Acc_r \approx Acc_u$), ensuring fairness while maintaining high $Acc_r$ and $Acc_u$ for utility. Tab. 4 shows that the Fine-tune method fails to unlearn bias, as $Acc_u$ remains higher than $Acc_r$, despite slightly higher $Acc_r$ compared to Retrain. FedCDP [65] and FedRecovery [61] exhibit catastrophic forgetting, with low $Acc_r$ and $Acc_u$, making them unsuitable for biased feature unlearning. In contrast, Ferrari demonstrates effective unlearning with similar $Acc_r$ and $Acc_u$, and maintains high overall accuracy, indicating a successful biased feature unlearning.

**Attention Map** Fig. 6b shows attention maps analyzing biased feature unlearning. The Baseline model predominantly focuses on the biased feature region (mouth) in both bias and unbias datasets, suggesting its significant impact on output prediction. However, Ferrari unlearned models redistribute attention across various facial regions in both datasets, similar to the Retrain model. This shift indicates reduced sensitivity to the biased feature, demonstrating successful unlearning. See Appendix A.3.2 for supplementary results.

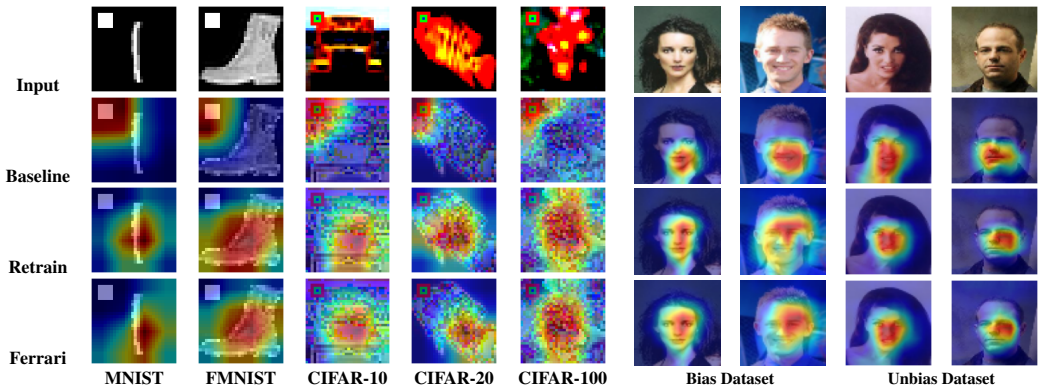
## 5.4 Computational Complexity

In Fig. 7, we evaluate the runtime performance and FLOPs metrics of each unlearning method to demonstrate the computational complexity. The Retrain method is expected to have the slowest runtime and highest FLOPs, while Fine-tune is fast but still slower than other methods.

Both FedCDP [65] and FedRecovery [61] demonstrate faster runtimes and lower FLOPs than the Fine-tune method, but they are still more computationally expensive than Ferrari. This is primarily due to the need to access training datasets from all clients and the computational expense of gradient residual calculations [61].

| Scenarios | Datasets | Unlearn Feature | | Accuracy (%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Baseline | Retrain | Fine-tune | FedCDP[65] | FedRecovery[61] | Ferrari(Ours) |
| Backdoor | MNIST | Backdoor pixel-pattern | $\mathcal{D}_r$ | 95.65 ±1.39 | 97.19 ±2.49 | **96.16 ±0.37** | 65.82 ±6.85 | 40.81 ±4.31 | 95.93 ±0.45 |
| | | | $\mathcal{D}_u$ | 97.43 ±3.69 | 0.00 ±0.00 | 72.64 ±0.24 | 69.37 ±0.83 | 53.72 ±3.14 | **0.11 ±0.01** |
| | FMNIST | | $\mathcal{D}_r$ | 91.07 ±0.54 | 93.85 ±1.08 | **94.36 ±1.98** | 68.46 ±3.39 | 42.93 ±2.50 | 92.83 ±0.61 |
| | | | $\mathcal{D}_u$ | 94.51 ±6.29 | 0.00 ±0.00 | 43.91 ±0.28 | 72.19 ±0.49 | 48.15 ±4.37 | **0.90 ±0.03** |
| | CIFAR-10 | | $\mathcal{D}_r$ | 87.63 ±1.16 | 91.12 ±1.60 | **92.02 ±3.15** | 54.91 ±6.91 | 27.49 ±4.96 | 89.91 ±0.95 |
| | | | $\mathcal{D}_u$ | 95.05 ±2.30 | 0.00 ±0.00 | 88.44 ±0.92 | 62.75 ±5.07 | 49.26 ±2.23 | **0.29 ±0.04** |
| | CIFAR-20 | | $\mathcal{D}_r$ | 75.06 ±6.41 | 81.91 ±4.68 | **82.67 ±1.32** | 55.67 ±6.35 | 23.76 ±2.17 | 78.29 ±3.12 |
| | | | $\mathcal{D}_u$ | 94.21 ±4.11 | 0.00 ±0.00 | 86.53 ±1.47 | 50.17 ±9.11 | 50.38 ±4.25 | **0.78 ±0.08** |
| | CIFAR-100 | | $\mathcal{D}_r$ | 54.14 ±3.96 | 73.54 ±5.70 | **73.66 ±6.57** | 34.62 ±2.24 | 15.62 ±7.78 | 69.57 ±3.81 |
| | | | $\mathcal{D}_u$ | 88.98 ±6.63 | 0.00 ±0.00 | 65.38 ±4.76 | 57.29 ±3.62 | 46.17 ±9.25 | **0.15 ±0.01** |
| | ImageNet | | $\mathcal{D}_r$ | 52.35 ±2.25 | 67.05 ±1.29 | **67.34 ±2.73** | 29.74 ±4.72 | 13.46 ±6.53 | 65.74 ±1.32 |
| | | | $\mathcal{D}_u$ | 83.16 ±3.74 | 0.00 ±0.00 | 71.48 ±3.69 | 62.39 ±3.05 | 54.92 ±5.59 | **0.09 ±0.02** |
| Biased | CMNIST | Color | $\mathcal{D}_r$ | 64.94 ±7.88 | 98.76 ±3.65 | 67.15 ±2.60 | 25.85 ±1.58 | 23.92 ±1.08 | **84.31 ±2.63** |
| | | | $\mathcal{D}_u$ | 98.88 ±4.90 | 98.44 ±1.90 | 97.95 ±1.13 | 30.17 ±4.69 | 27.64 ±9.37 | **84.62 ±3.59** |
| | CelebA | Mouth | $\mathcal{D}_r$ | 79.46 ±2.09 | 96.47 ±6.15 | 84.45 ±1.48 | 14.29 ±0.81 | 16.34 ±3.43 | **94.18 ±3.08** |
| | | | $\mathcal{D}_u$ | 96.38 ±3.87 | 96.11 ±2.17 | 94.23 ±0.66 | 21.58 ±3.48 | 25.72 ±8.02 | **94.79 ±1.48** |

Table 4: The accuracy of $\mathcal{D}_r$ and $\mathcal{D}_u$ for each unlearning method across different unlearning scenarios.




(a) Backdoor Feature Unlearning (b) Biased Feature Unlearning

Figure 6: The attention map of each unlearning method across different unlearning scenarios.

In contrast, Ferrari has the lowest computational complexity, with the fastest runtime and lowest FLOPs. It only requires access to the local dataset of the unlearn client and achieves feature unlearning by minimizing feature sensitivity within a single epoch.

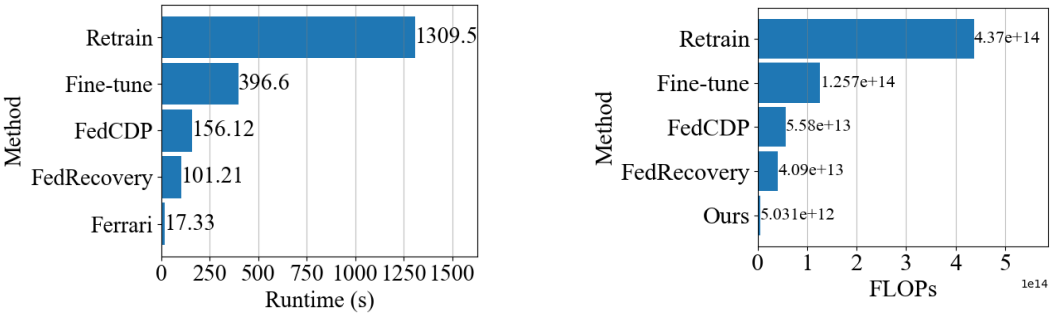
### 5.5 Ablation Study and Hyper-parameter Analysis

We conduct an ablation study to analyze how Non-Lipschitz affects the effectiveness of our proposed Ferrari and hyper-parameter analysis of Gaussian noise level ($\sigma$) and number of $\mathcal{D}_u$ in Fig. 8.

**Non-Lipschitz** We evaluate the unlearning performance by removing the denominator in Eq. 6, calling this the Non-Lipschitz method, as shown in Fig. 8a. The results indicate catastrophic forgetting: $\mathcal{D}_r$ accuracy drops below 10%, and the unlearned model misclassifies all inputs into a single random class, rendering it useless. This stems from the unbounded loss function in the non-Lipschitz method, unlike the bounded Lipschitz constant in Eq. 6, which provides a theoretical guarantee (see Sec. 4.3). Refer to Appendix A.4 for a detailed analysis of Lipschitz and Non-Lipschitz loss functions.

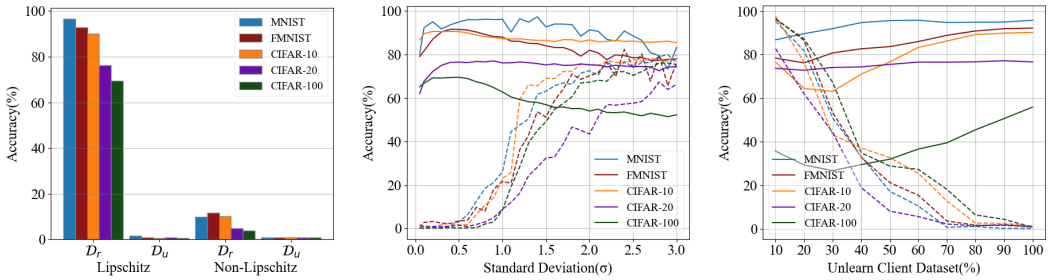
**Gaussian Noise** The effectiveness of Ferrari is significantly influenced by injected Gaussian noise. Fig. 8b shows the accuracy of $\mathcal{D}_r$ and $\mathcal{D}_u$ across different $\sigma$ levels. In the $0.05 \leq \sigma \leq 1.0$ range, $\mathcal{D}_r$ accuracy stays high and $\mathcal{D}_u$ accuracy remains low, indicating a balance. Thus, we implement $\sigma$ values between 0.05 and 1.0 for a balanced accuracy across $\mathcal{D}_r$ and $\mathcal{D}_u$.

**Number of Unlearn Dataset** Our analysis illustrated in Fig. 8c, demonstrates that Ferrari remains effective with partial $\mathcal{D}_u$ from $C_u$ for feature unlearning (*i.e.,* data lost). Using 70% of $\mathcal{D}_u$ yields comparable accuracy to using the full (*i.e.,* 100%) dataset, highlighting the method's flexibility even with partial data.




(a) Runtime(s) (b) FLOPs

Figure 7: Computational complexity analysis comparing the runtime(s) and FLOPs for each unlearning method.




(a) Non-Lipschitz (b) Gaussian noise level($\sigma$) (c) Number of $\mathcal{D}_u$

Figure 8: Ablation and hyper-parameter analysis on Ferrari backdoor feature unlearning. Solid line: $\mathcal{D}_r$; dashed line: $\mathcal{D}_u$.

## 6 Conclusion

This paper introduces Ferrari, a federated feature unlearning framework designed to efficiently remove sensitive, backdoor, and biased features without extensive retraining. Leveraging Lipschitz continuity, Ferrari reduces model sensitivity to specific features, ensuring robust and fair models. Uniquely, it requires participation only from the client requesting unlearning, preserving privacy and practicality in FL environments. Experimental results and theoretical analysis demonstrate Ferrari's effectiveness across various data domains, addressing the crucial need for feature-level unlearning in federated learning. This method can serve as a technical solution to meet regulatory requirements for data deletion while maintaining model performance, offering significant value to clients by securing their "right to be forgotten" and preventing potential privacy leakage.

### 6.1 Limitation and Future Work

The proposed federated feature unlearning method works effectively using only the unlearning client's local data, making it well-suited for real-world scenarios. However, for optimal results, access to the full dataset is required. As demonstrated in Section 5.5, using 70% of the data yields comparable performance, but significant data reduction diminishes effectiveness. Future research should focus on developing methods that require only a small portion of the client's data and expanding the approach beyond classification models to include for example, generative models. Additionally, enhancements such as advanced perturbation techniques and integration with privacy-preserving methods should be explored.

## Acknowledgement

This research is supported by the Fundamental Research Grant Scheme (FRGS/1/2024/ICT02/UM/01/1), awarded by the Ministry of Higher Education, Malaysia.

## References


[1] J. Konečnỳ, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[4] E. L. Harding, J. J. Vanto, R. Clark, L. Hannah Ji, and S. C. Ainsworth, "Understanding the scope and impact of the california consumer privacy act of 2018," *Journal of Data Protection & Privacy*, vol. 2, no. 3, pp. 234–253, 2019.

[5] T. Che, Y. Zhou, Z. Zhang, L. Lyu, J. Liu, D. Yan, D. Dou, and J. Huan, "Fast federated machine unlearning with nonlinear functional theory," in *International conference on machine learning*, pp. 4241–4268, PMLR, 2023.

[6] Z. Liu, Y. Jiang, J. Shen, M. Peng, K.-Y. Lam, X. Yuan, and X. Liu, "A survey on federated unlearning: Challenges, methods, and future directions," 2024.

[7] N. Romandini, A. Mora, C. Mazzocca, R. Montanari, and P. Bellavista, "Federated unlearning: A survey on methods, design guidelines, and evaluation metrics," 2024.

[8] J. Yang and Y. Zhao, "A survey of federated unlearning: A taxonomy, challenges and future directions," 2023.

[9] A. Warnecke, L. Pirch, C. Wressnegger, and K. Rieck, "Machine unlearning of features and labels," in *Proc. of the 30th Network and Distributed System Security (NDSS)*, 2023.

[10] T. Guo, S. Guo, J. Zhang, W. Xu, and J. Wang, "Efficient attribute unlearning: Towards selective removal of input attributes from feature representations," 2022.

[11] M. Fredrikson, E. Lantz, S. Jha, S. M. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," *Proceedings of the USENIX Security Symposium. UNIX Security Symposium*, vol. 2014, pp. 17–32, 2014.

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

[13] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 250–258, 2020.

[14] S. Mehnaz, S. V. Dibbo, E. Kabir, N. Li, and E. Bertino, "Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models," in *USENIX Security Symposium*, 2022.

[15] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (S. Chiappa and R. Calandra, eds.), vol. 108 of *Proceedings of Machine Learning Research*, pp. 2938–2948, PMLR, 26–28 Aug 2020.

[16] T. D. Nguyen, T. Nguyen, P. L. Nguyen, H. H. Pham, K. D. Doan, and K.-S. Wong, "Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107166, 2024.

[17] H. Li, C. Wu, S. Zhu, and Z. Zheng, "Learning to backdoor federated learning," in *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*, 2023.

[18] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[19] D. Pessach and E. Shmueli, "A review on fairness in machine learning," *ACM Comput. Surv.*, vol. 55, feb 2022.

[20] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, jul 2021.

[21] S. Sagawa*, P. W. Koh*, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks," in *International Conference on Learning Representations*, 2020.

[22] S. Seo, J.-Y. Lee, and B. Han, "Unsupervised learning of debiased representations with pseudo-attributes," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16721–16730, 2022.

[23] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *2015 IEEE symposium on security and privacy*, pp. 463–480, IEEE, 2015.

[24] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang, "When machine unlearning jeopardizes privacy," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, (New York, NY, USA), p. 896–911, Association for Computing Machinery, 2021.

[25] S. Garg, S. Goldwasser, and P. N. Vasudevan, "Formalizing data deletion in the context of the right to be forgotten," in *39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings*, vol. 12105 of *Lecture Notes in Computer Science*, Springer, 2020.

[26] A. Thudi, H. Jia, I. Shumailov, and N. Papernot, "On the necessity of auditable algorithmic definitions for machine unlearning," in *USENIX Security Symposium*, 2021.

[27] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen, "A survey of machine unlearning," 2022.

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

[29] H. Yan, X. Li, Z. Guo, H. Li, F. Li, and X. Lin, "Arcane: An efficient architecture for exact machine unlearning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22* (L. D. Raedt, ed.), pp. 4006–4013, International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[30] J. Xu, Z. Wu, C. Wang, and X. Jia, "Machine unlearning: Solutions and challenges," *ArXiv*, vol. abs/2308.07061, 2023.

[31] H. Xu, T. Zhu, L. Zhang, W. Zhou, and P. S. Yu, "Machine unlearning: A survey," *ACM Comput. Surv.*, vol. 56, aug 2023.

[32] L. Graves, V. Nagisetty, and V. Ganesh, "Amnesiac machine learning," in *AAAI Conference on Artificial Intelligence*, 2020.

[33] J. Kim and S. S. Woo, "Efficient two-stage model retraining for machine unlearning," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4360–4368, 2022.

[34] S. Lee and S. S. Woo, "Undo: Effective and accurate unlearning method for deep neural networks," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 4043–4047, 2023.

[35] M. Chen, W. Gao, G. Liu, K. Peng, and C. Wang, "Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7766–7775, 2023.

[36] T. Shibata, G. Irie, D. Ikami, and Y. Mitsuzumi, "Learning with selective forgetting," in *International Joint Conference on Artificial Intelligence*, 2021.

[37] H. Huang, X. Ma, S. M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable," in *International Conference on Learning Representations*, 2021.

[38] J. Z. Di, J. Douglas, J. Acharya, G. Kamath, and A. Sekhari, "Hidden poison: Machine unlearning enables camouflaged poisoning attacks," in *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*, 2022.

[39] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, "Fast yet effective machine unlearning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2023.

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

[41] X. Zhang, J. Wang, N. Cheng, Y. Sun, C. Zhang, and J. Xiao, "Machine unlearning methodology based on stochastic teacher network," in *Advanced Data Mining and Applications* (X. Yang, H. Suhartanto, G. Wang, B. Wang, J. Jiang, B. Li, H. Zhu, and N. Cui, eds.), (Cham), pp. 250–261, Springer Nature Switzerland, 2023.

[42] M. Kurmanji, P. Triantafillou, J. Hayes, and E. Triantafillou, "Towards unbounded machine unlearning," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[43] Y. Jung, I. Cho, S.-H. Hsu, and J. Hockenmaier, "Attack and reset for unlearning: Exploiting adversarial noise toward machine unlearning through parameter re-initialization," 2024.

[44] T. Hoang, S. Rana, S. Gupta, and S. Venkatesh, "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 (WACV)*, pp. 4819–4828, January 2024.

[45] A. Abbasi, C. Thrash, E. Akbari, D. Zhang, and S. Kolouri, "Covarnav: Machine unlearning via model inversion and covariance navigation," 2023.

[46] D. Choi and D. Na, "Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems," 2023.

[47] S. Goel, A. Prabhu, A. Sanyal, S.-N. Lim, P. Torr, and P. Kumaraguru, "Towards adversarial evaluations for inexact machine unlearning," 2023.

[48] A. Golatkar, A. Achille, and S. Soatto, "Eternal sunshine of the spotless net: Selective forgetting in deep networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9301–9309, 2020.

[49] A. Golatkar, A. Achille, and S. Soatto, "Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 383–398, Springer International Publishing, 2020.

[50] A. Golatkar, A. Achille, A. Ravichandran, M. Polito, and S. Soatto, "Mixed-privacy forgetting in deep networks," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 792–801, 2021.

[51] C. Guo, T. Goldstein, A. Hannun, and L. Van Der Maaten, "Certified data removal from machine learning models," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842, PMLR, 13–18 Jul 2020.

[52] J. Foster, S. Schoepf, and A. Brintrup, "Fast machine unlearning without retraining through selective synaptic dampening," 2023.

[53] C. Guo, T. Goldstein, A. Hannun, and L. Van Der Maaten, "Certified data removal from machine learning models," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842, PMLR, 13–18 Jul 2020.

[54] C. Wu, S. Zhu, and P. Mitra, "Federated unlearning with knowledge distillation," 2022.

[55] G. Liu, X. Ma, Y. Yang, C. Wang, and J. Liu, "Federaser: Enabling efficient client-level data removal from federated learning models," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pp. 1–10, 2021.

[56] W. Yuan, H. Yin, F. Wu, S. Zhang, T. He, and H. Wang, "Federated unlearning for on-device recommendation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, (New York, NY, USA), p. 393–401, Association for Computing Machinery, 2023.

[57] X. Cao, J. Jia, Z. Zhang, and N. Z. Gong, "Fedrecover: Recovering from poisoning attacks in federated learning using historical information," in *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1366–1383, 2023.

[58] X. Gao, X. Ma, J. Wang, Y. Sun, B. Li, S. Ji, P. Cheng, and J. Chen, "Verifi: Towards verifiable federated unlearning," 2022.

[59] G. Ye, T. Chen, Q. V. Hung Nguyen, and H. Yin, "Heterogeneous decentralised machine unlearning with seed model distillation," *CAAI Transactions on Intelligence Technology*, 2024.

[60] N. Su and B. Li, "Asynchronous federated unlearning," in *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, pp. 1–10, 2023.

[61] L. Zhang, T. Zhu, H. Zhang, P. Xiong, and W. Zhou, "Fedrecovery: Differentially private machine unlearning for federated learning frameworks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4732–4746, 2023.

[62] A. Halimi, S. Kadhe, A. Rawat, and N. Baracaldo, "Federated unlearning: How to efficiently erase a client in fl?," 2023.

[63] G. Li, L. Shen, Y. Sun, Y. Hu, H. Hu, and D. Tao, "Subspace based federated unlearning," 2023.

[64] M. Alam, H. Lamri, and M. Maniatakos, "Get rid of your trail: Remotely erasing backdoors in federated learning," 2023.

[65] J. Wang, S. Guo, X. Xie, and H. Qi, "Federated unlearning via class-discriminative pruning," in *Proceedings of the ACM Web Conference 2022*, WWW '22, (New York, NY, USA), p. 622–632, Association for Computing Machinery, 2022.

[66] Y. Zhao, P. Wang, H. Qi, J. Huang, Z. Wei, and Q. Zhang, "Federated unlearning with momentum degradation," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[67] Y. Liu, L. Xu, X. Yuan, C. Wang, and B. Li, "The right to be forgotten in federated learning: An efficient realization with rapid retraining," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pp. 1749–1758, 2022.

[68] A. Dhasade, Y. Ding, S. Guo, A. marie Kermarrec, M. D. Vos, and L. Wu, "Quickdrop: Efficient federated unlearning by integrated dataset distillation," 2023.

[69] P. Wang, Z. Yan, M. S. Obaidat, Z. Yuan, L. Yang, J. Zhang, Z. Wei, and Q. Zhang, "Edge caching with federated unlearning for low-latency v2x communications," *IEEE Communications Magazine*, pp. 1–7, 2023.

[70] H. Xia, S. Xu, J. Pei, R. Zhang, Z. Yu, W. Zou, L. Wang, and C. Liu, "Fedme2: Memory evaluation & erase promoting federated unlearning in dtmn," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3573–3588, 2023.

[71] S. Moon, S. Cho, and D. Kim, "Feature unlearning for pre-trained gans and vaes," in *AAAI Conference on Artificial Intelligence*, 2023.

[72] S. Bae, S. Kim, H. Jung, and W. Lim, "Gradient surgery for one-shot unlearning on generative model," 2023.

[73] Anonymous, "Machine unlearning for image-to-image generative models," in *The Twelfth International Conference on Learning Representations*, 2024.

[74] W. Wang, H. Bai, J. tse Huang, Y. Wan, Y. Yuan, H. Qiu, N. Peng, and M. R. Lyu, "New job, new gender? measuring the social bias in image generation models," 2024.

[75] Y. Yao, X. Xu, and Y. Liu, "Large language model unlearning," 2023.

[76] C. Yu, S. Jeoung, A. Kasi, P. Yu, and H. Ji, "Unlearning bias in language models by partitioning gradients," in *Findings of the Association for Computational Linguistics: ACL 2023* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 6032–6048, Association for Computational Linguistics, July 2023.

[77] J. Chen and D. Yang, "Unlearn what you want to forget: Efficient unlearning for LLMs," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 12041–12052, Association for Computational Linguistics, Dec. 2023.

[78] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," *arXiv preprint arXiv:1705.10941*, 2017.

[79] A. M. Oberman and J. Calder, "Lipschitz regularized deep neural networks converge and generalize," *CoRR*, vol. abs/1808.09540, 2018.

[80] G. Khromov and S. P. Singh, "Some fundemental aspects about lipschitz continuity of neural networks," 2023.

[81] M. Usama and D. E. Chang, "Towards robust neural networks with lipschitz continuity," in *Digital Forensics and Watermarking* (C. D. Yoo, Y.-Q. Shi, H. J. Kim, A. Piva, and G. Kim, eds.), (Cham), pp. 373–389, Springer International Publishing, 2019.

[82] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," in *International Conference on Learning Representations*, 2018.

[83] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," 2017.

[84] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.

[85] Wenruliu, "Adult income dataset." Kaggle, 2024.

[86] M. Akturk, "Diabetes dataset." Kaggle, 2024.

[87] Y. Djebrouni, N. Benarba, O. Touat, P. De Rosa, S. Bouchenak, A. Bonifati, P. Felber, V. Marangozova, and V. Schiavoni, "Bias mitigation in federated learning for edge computing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 7, jan 2024.

[88] R. Chen, J. Yang, H. Xiong, J. Bai, T. Hu, J. Hao, Y. FENG, J. T. Zhou, J. Wu, and Z. Liu, "Fast model debias with machine unlearning," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[89] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[90] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[91] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[92] A. Krizhevsky, "Learning multiple layers of features from tiny images," Toronto, ON, Canada, 2009.

[93] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[94] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019.

[95] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (D. Lin, Y. Matsumoto, and R. Mihalcea, eds.), (Portland, Oregon, USA), pp. 142–150, Association for Computational Linguistics, June 2011.

[96] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

# A Appendix

## A.1 Proof of Theorem 1

As illustrated in Sec. 3.2, it is hard to build the unlearned data $x^u$ for the feature unlearning since adding the perturbation may influence the model accuracy seriously. Suppose the feature is successfully removed when the norm of perturbation is larger than $C$. We define the utility loss $\ell_1$ with unlearning feature successfully:

$$\ell_1 = \min_{\|\delta_{\mathcal{F}}\| \geq C} \mathbb{E}_{(x,y)\in\mathcal{D}} \min_{\theta} \ell\big(f_\theta(x+\delta_{\mathcal{F}}), y\big) \tag{10}$$

And we define the maximum utility loss with the norm perturbation less than $C$ as:

$$\ell_2 = \max_{\|\delta_{\mathcal{F}}\| \leq C} \mathbb{E}_{(x,y)\in\mathcal{D}} \min_{\theta} \ell\big(f_\theta(x+\delta_{\mathcal{F}}), y\big) \tag{11}$$

**Assumption 3.** *Assume $\ell_2 \leq \ell_1$*

Assumption 3 elucidates that the utility loss associated with a perturbation norm less than $C$ is smaller than the utility loss when the perturbation norm is greater than $C$. This assumption is logical, as larger perturbations would naturally lead to greater utility loss.

**Assumption 4.** *Suppose the federated model achieves zero training loss.*

We have the following theorem to elucidate the relation between feature sensitivity removing via Alg. 1 and exact unlearning (see proof in Appendix).

**Theorem 2.** *If Assumption 3 and 4 hold, the utility loss of unlearned model obtained by Alg. 1 is less than the utility loss with unlearning successfully, i.e.,,*

$$\ell_u \leq \ell_1, \tag{12}$$

*where $\ell_u = \mathbb{E}_{(x,y)\in\mathcal{D}}\big(\ell(f_{\theta^u}(x), y)$*

*Proof.* When the unlearning happens during the federated training, the unlearning clients would also optimize the training loss and feature sensitivity simultaneously. Specifically, the optimization process could be written as:

$$\theta_u = \arg\min_{\theta} \mathbb{E}_{(x,y)\in\mathcal{D}}\big(\ell(f_\theta(x), y) + \lambda \mathbb{E}_{\delta_{\mathcal{F}}} \frac{\|f_\theta(x) - f_\theta(x+\delta_{\mathcal{F}})\|_2}{\|\delta_{\mathcal{F}}\|_2}\big),$$

where $\lambda \leq \|\delta_{\mathcal{F}}\| \leq C$ is one coefficient. Without loss of generality, we assume the $\ell(f_\theta(x), y) = \|f_\theta(x) - y)\|$. Denote

$$\Theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\in\mathcal{D}} \ell(f_\theta(x), y).$$

If Assumption 4 holds, then $f_{\theta^*}(x) = y$ for any $\theta^* \in \Theta^*$. Therefore, for any $\lambda \leq \|\delta_{\mathcal{F}}\| \leq C$ such that

$$\begin{aligned}
&\mathbb{E}_{(x,y)\in\mathcal{D}}\big(\ell(f_{\theta^*}(x), y) + \lambda \mathbb{E}_{\lambda \leq \|\delta_{\mathcal{F}}\| \leq C} \frac{\|f_\theta(x) - f_{\theta^*}(x+\delta_{\mathcal{F}})\|_2}{\|\delta_{\mathcal{F}}\|_2}\big) \\
&= \lambda \mathbb{E}_{(x,y)\in\mathcal{D}} \mathbb{E}_{\lambda \leq \|\delta_{\mathcal{F}}\| \leq C} \frac{\|y - f_{\theta^*}(x+\delta_{\mathcal{F}})\|_2}{\|\delta_{\mathcal{F}}\|_2} \\
&\leq \mathbb{E}_{(x,y)\in\mathcal{D}} \mathbb{E}_{\lambda \leq \|\delta_{\mathcal{F}}\| \leq C} \|y - f_{\theta^*}(x+\delta_{\mathcal{F}})\|_2.
\end{aligned} \tag{13}$$

Therefore, we further obtain:

$$
\begin{aligned}
\ell_u &= \min_{\theta\in\mathbb{R}^d} \mathbb{E}_{(x,y)\in\mathcal{D}}\big(\ell(f_\theta(x),y) + \lambda\mathbb{E}_{\lambda\le\|\delta_\mathcal{F}\|\le C}\frac{\|f_\theta(x)-f_\theta(x+\delta_\mathcal{F})\|_2}{\|\delta_\mathcal{F}\|_2}\big) \\
&\le \min_{\theta\in\Theta^*} \mathbb{E}_{(x,y)\in\mathcal{D}}\big(\ell(f_\theta(x),y) + \lambda\mathbb{E}_{\lambda\le\|\delta_\mathcal{F}\|\le C}\frac{\|f_\theta(x)-f_\theta(x+\delta_\mathcal{F})\|_2}{\|\delta_\mathcal{F}\|_2}\big) \\
&\le \min_{\theta\in\Theta^*} \mathbb{E}_{(x,y)\in\mathcal{D}}\mathbb{E}_{\lambda\le\|\delta_\mathcal{F}\|\le C}\|y-f_{\theta^*}(x+\delta_\mathcal{F})\|_2 \\
&\le \mathbb{E}_{(x,y)\in\mathcal{D}}\mathbb{E}_{\lambda\le\|\delta_\mathcal{F}\|\le C}\min_{\theta\in\Theta^*}\|y-f_{\theta^*}(x+\delta_\mathcal{F})\|_2 \\
&= \mathbb{E}_{\lambda\le\|\delta_\mathcal{F}\|\le C}\mathbb{E}_{(x,y)\in\mathcal{D}}\min_{\theta\in\Theta^*}\|y-f_{\theta^*}(x+\delta_\mathcal{F})\|_2 \\
&\le \max_{\lambda\le\|\delta_\mathcal{F}\|\le C}\mathbb{E}_{(x,y)\in\mathcal{D}}\min_{\theta\in\mathbb{R}^d}\|y-f_{\theta^*}(x+\delta_\mathcal{F})\|_2 \\
&\le \max_{\|\delta_\mathcal{F}\|\le C}\mathbb{E}_{(x,y)\in\mathcal{D}}\min_{\theta\in\mathbb{R}^d}\|y-f_{\theta^*}(x+\delta_\mathcal{F})\|_2 \\
&= \ell_2,
\end{aligned}
\tag{14}
$$

According to Assumption 3, we have $\ell_u \le \ell_1$

□

### A.2 Experimental Setup

**Datasets** *MNIST* [89]: Both the *MNIST* [89] and *Fashion-MNIST(FMNIST)* [91] datasets contain images of handwritten digits and attire, respectively. Each dataset comprises 60,000 training examples and 10,000 test examples. In both datasets, each example is represented as a single-channel image with dimensions of 28×28 pixels, categorized into one of 10 classes. Additionally, the *Colored-MNIST(CMNIST)* [89] dataset, an extension of the original MNIST, introduces color into the digits of each example. Consequently, images in the Colored MNIST dataset are represented in three channels. *CIFAR* [92]: The *CIFAR-10* [92] dataset comprises 60,000 images, each with dimensions of 32×32 pixels and three color channels, distributed across 10 classes. This dataset includes 6,000 images per class and is partitioned into 50,000 training examples and 10,000 test examples. Similarly, the *CIFAR-100* [92] dataset shares the same image dimensions and structure as *CIFAR-10* but extends to 100 classes, with each class containing 600 images. Within each class, there are 500 training images and 100 test images. Moreover, *CIFAR-100* organizes its 100 classes into 20 superclasses, forming the *CIFAR-20 dataset* [92]. *CelebA* [84]: A face recognition dataset featuring 40 attributes such as gender and facial characteristics, comprising 162,770 training examples and 19,962 test examples. This study will focus on utilizing the *CelebA* [84] dataset primarily for gender classification tasks. ImageNet [93]: A large-scale image dataset which contains 1.2 million training samples across 1,000 categories.

*Adult Census Income (Adult)* [85] includes 48, 842 records with 14 attributes such as age, gender, education, marital status, etc. The classification task of this dataset is to predict if a person earns over $50K a year based on the census attributes. We then consider marital status as the sensitive feature that aim to unlearn in this study. *Diabetes* [86] includes 768 personal health records of females at least 21 years old with 8 attributes such as blood pressure, insulin level, age and etc. The classification task of this dataset is to predict if a person has diabetes. We then consider number of pregnancies as the sensitive feature that aim to unlearn in this study.

The IMDB movie reviews dataset [95] is widely used for binary sentiment analysis, where the task is to determine whether a review expresses a positive or negative sentiment. It comprises 50,000 movie reviews, each labeled as either positive or negative. In this study, we focus on unlearning the influence of specific sensitive features, particularly the names of celebrities. Each client's local dataset includes names of specific celebrities, which are treated as sensitive features for this analysis.

**Baselines** The baseline methods in this study:

*Baseline*: Original model before unlearning.

*Retrain*: In scenarios involving sensitive feature unlearning, the retrained model was simply trained using a dataset where Gaussian noise was applied to the unlearned feature region. This approach

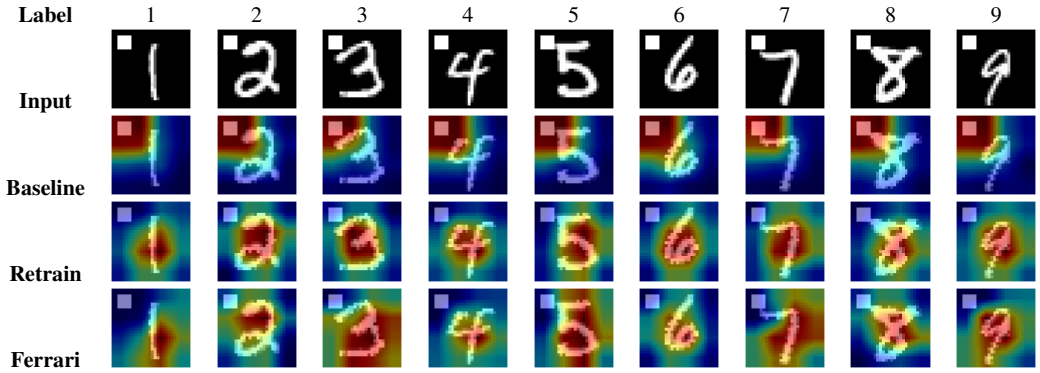



Figure 9: MNIST

may lead to performance deterioration, as discussed in Sec. 3.2. For backdoor feature unlearning scenarios, the retrained model was trained using the retain dataset $\mathcal{D}_r$, also referred to as the clean dataset. In biased feature unlearning scenarios, the retrained model was trained using a combination of 50% from each of the retain dataset $\mathcal{D}_r$ (bias dataset) and the unlearn client local dataset $\mathcal{D}_u$ (unbias dataset). This ensures fairness in the model's performance across both datasets.

*Fine-tune*: The baseline model is fine-tuned using the retained dataset $\mathcal{D}_r$ for 5 epochs.

*Class-Discriminative Pruning(FedCDP)* [65]: A FU framework that achieves class unlearning by utilizing Term Frequency-Inverse Document Frequency (TF-IDF) guided channel pruning, which selectively removes the most discriminative channels related to the target category and followed by fine-tuning without retraining from scratch.

*FedRecovery* [61]: A FU framework that achieves client unlearning by removing the influence of a client's data from the global model using a differentially private machine unlearning algorithm that leverages historical gradient submissions without the need for retraining.

### A.3 Attention Map

In this section, we provide additional results from attention map analysis based on GradCAM [96] for backdoor feature unlearning (refer to Sec. A.3.1) and biased feature unlearning (refer to Sec. A.3.2)

#### A.3.1 Backdoor Feature Unlearning

Attention map analysis for backdoor samples across model iterations of baseline, retrain, and unlearn model using our proposed Ferrari method on MNIST (Fig. 9), FMNIST (Fig. 10), CIFAR-10 (Fig. 11), CIFAR-20 (Fig. 12) and CIFAR-100 (Fig. 13) datasets.

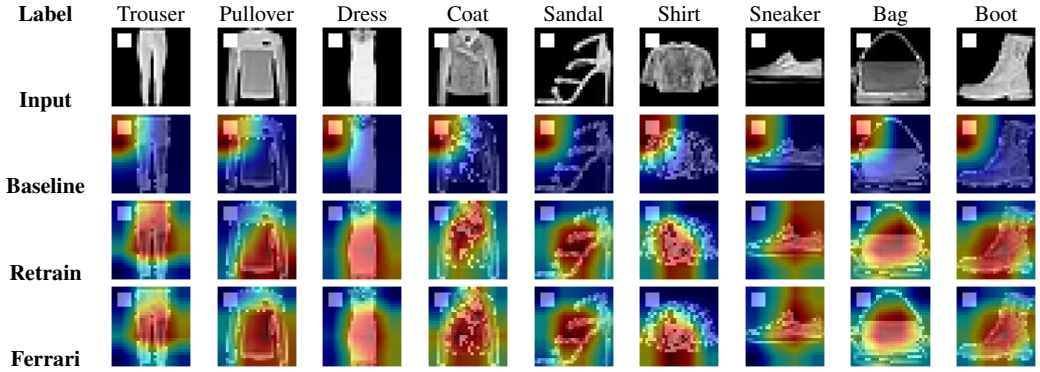



Figure 10: FMNIST

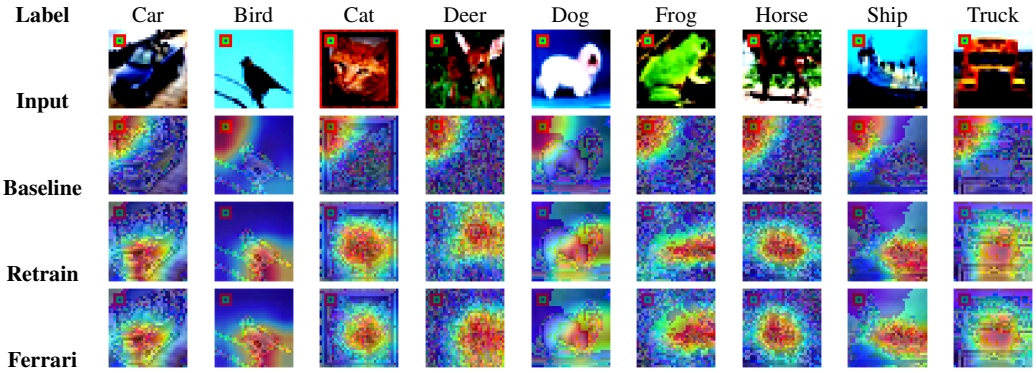



Figure 11: CIFAR-10

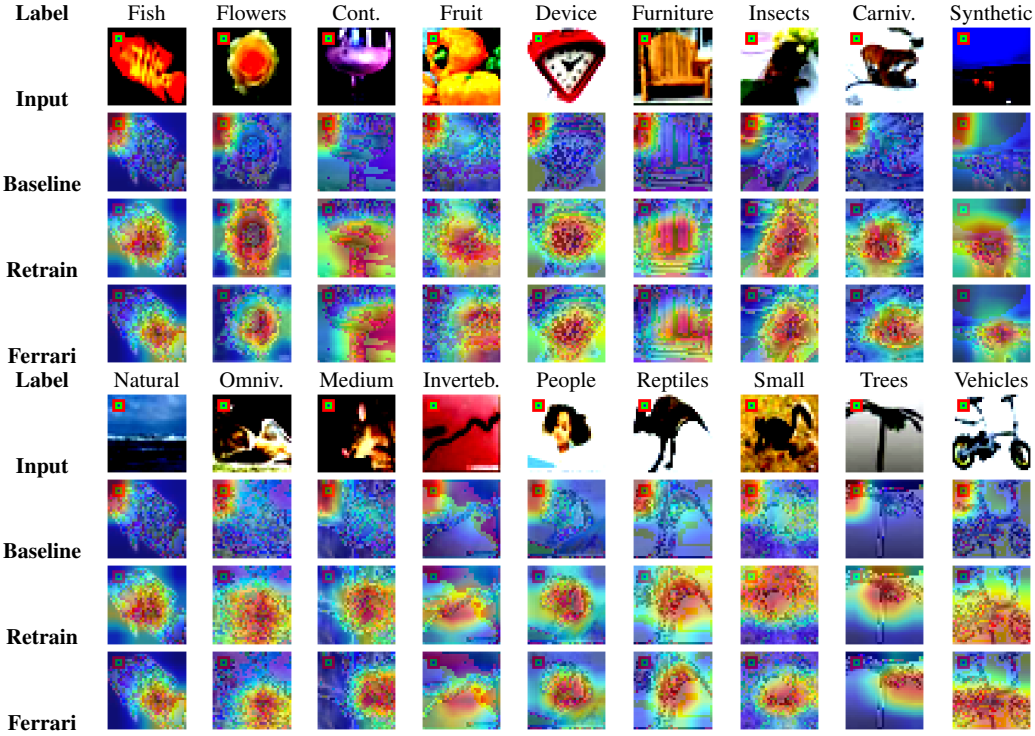
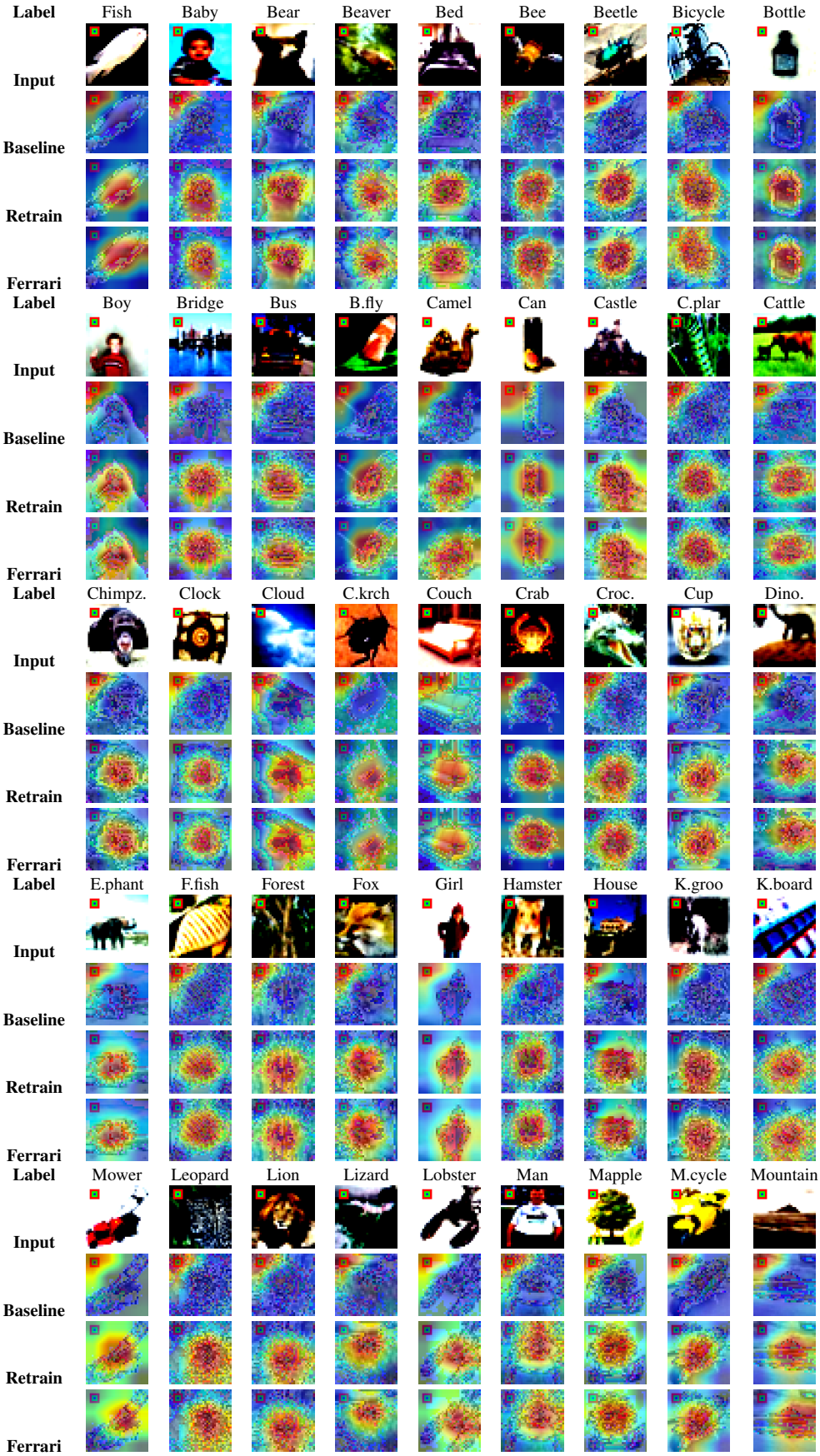



Figure 12: CIFAR-20




Continued on next page

Continued from previous page

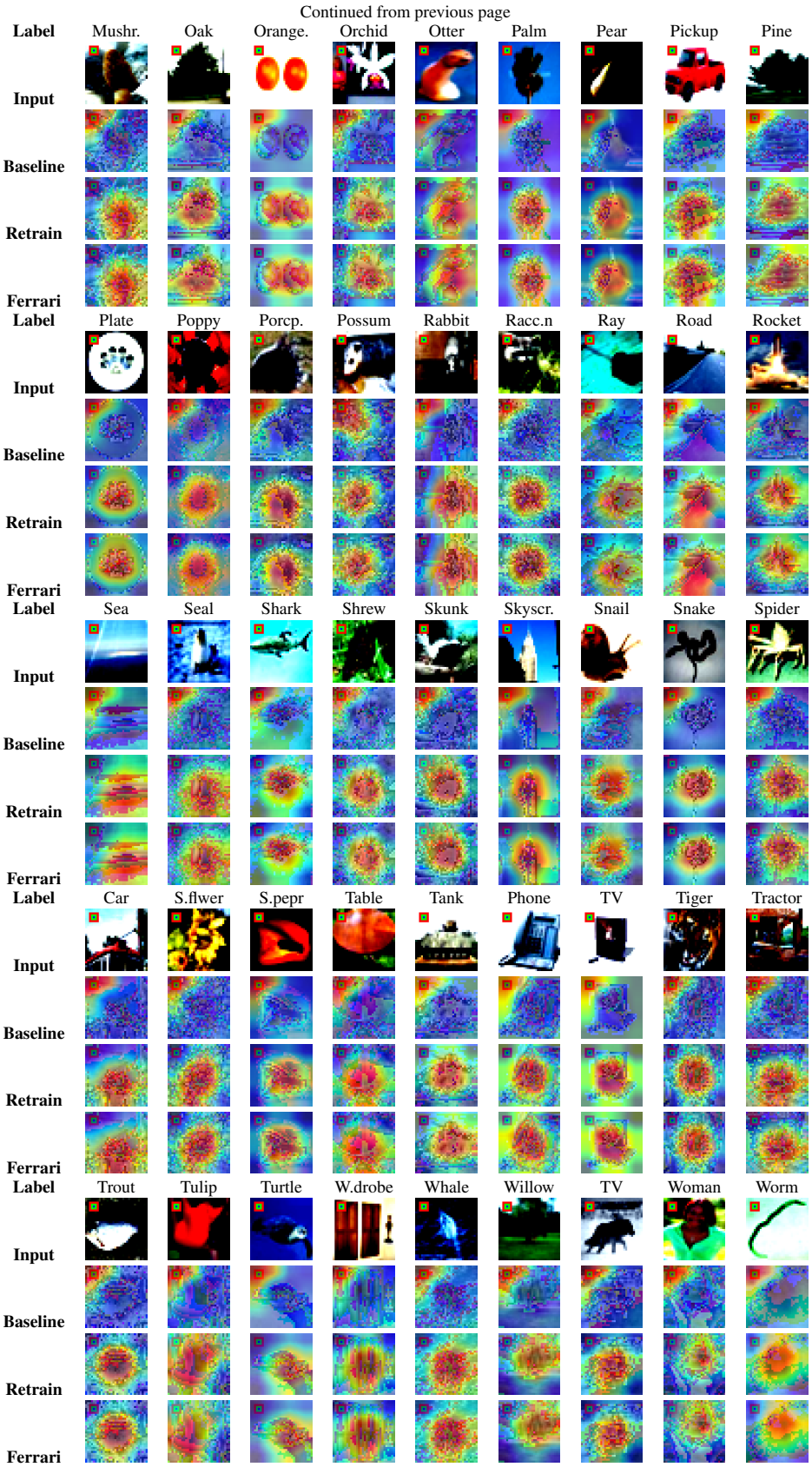
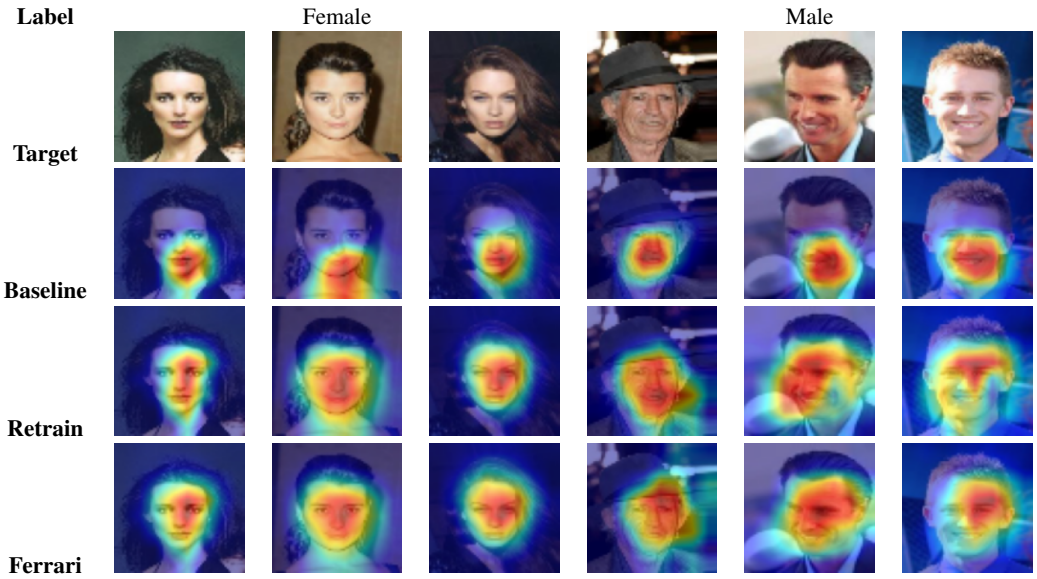
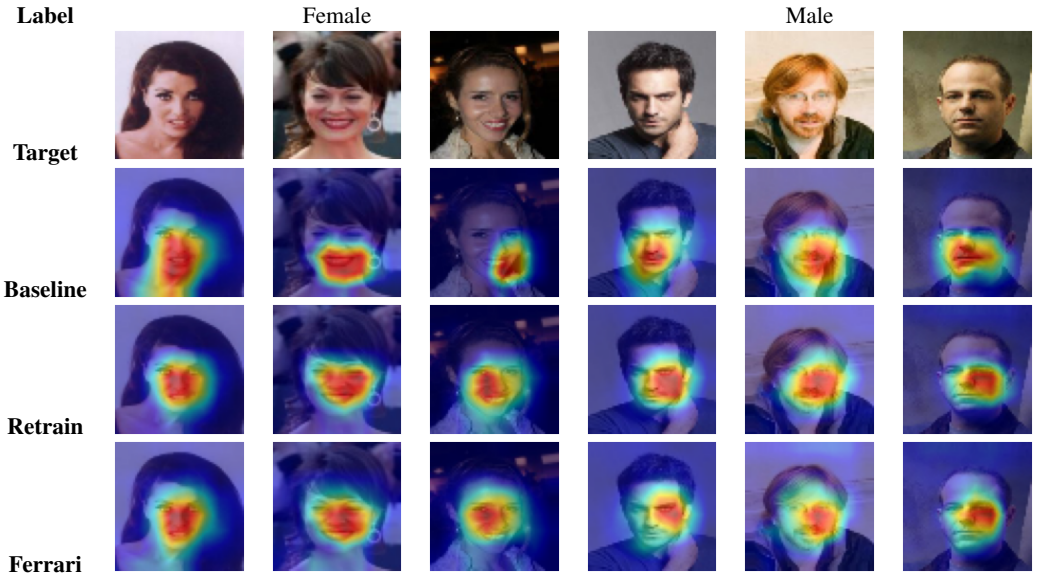



Figure 13: CIFAR-100

### A.3.2 Biased Feature Unlearning




(a) Bias Dataset




(b) Unbias Dataset

Figure 14: Attention map analysis for bias and unbias samples across model iterations of baseline, retrain, and unlearn model using our proposed Ferrari to unlearn 'mouth' on CelebA dataset.

## A.4 Lipschitz and Non-Lipschitz Loss Analysis

In this section, we evaluate the Lipschitz loss function and its effectiveness in optimizing feature sensitivity, as described in Eq. 6. We also examine a variant without the denominator, termed the Non-Lipschitz loss, as illustrated in Fig. 15.

The results indicate that models optimized using the Non-Lipschitz loss exhibit fluctuations across batches. This is due to the unbounded nature of the optimization process, leading to useless models. Fig. 8a further illustrates this issue, showing instances of catastrophic forgetting.

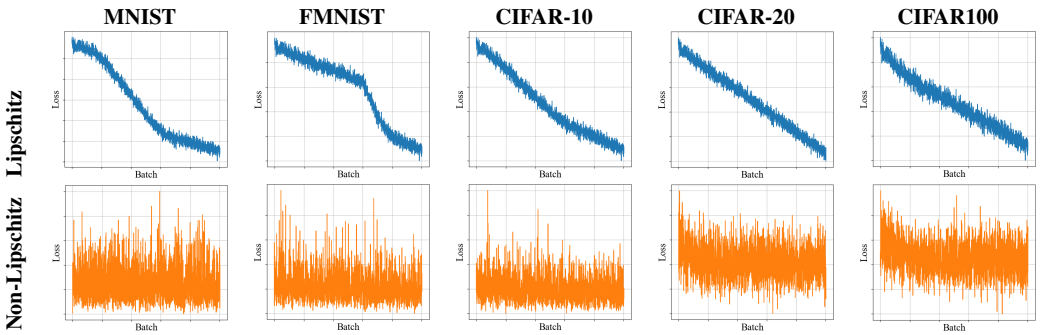
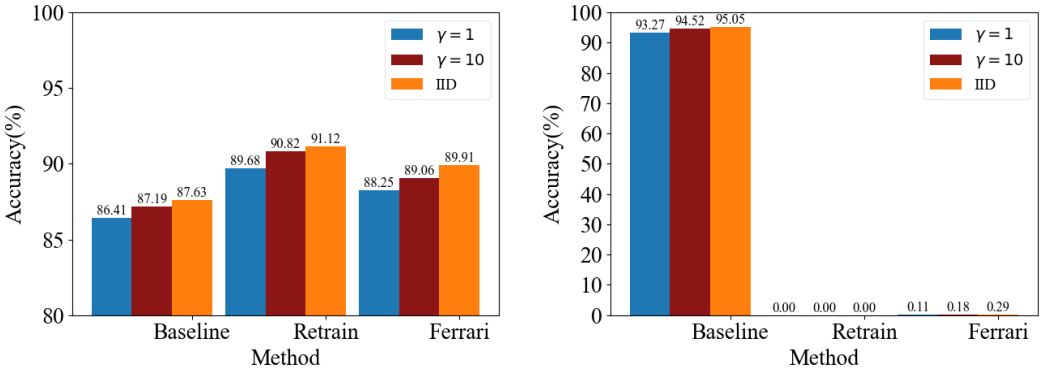



Figure 15: Lipschitz and Non-Lipschitz loss analysis on backdoor feature unlearning.




(a) $\mathcal{D}_r$ (b) $\mathcal{D}_u$

Figure 16: Non-IID analysis on the CIFAR-10 dataset using our proposed Ferrari framework, compared to Baseline and Retrain methods, for retain client dataset $\mathcal{D}_r$ and unlearn client dataset $\mathcal{D}_u$ accuracy in backdoor feature unlearning.

Conversely, models optimized with the Lipschitz loss demonstrate a steady reduction in feature sensitivity over batches. This bounded optimization provided by Lipschitz bound helps in effectively unlearning target features while preserving model utility, as theoretically guaranteed (see Section Sec. 4.3).

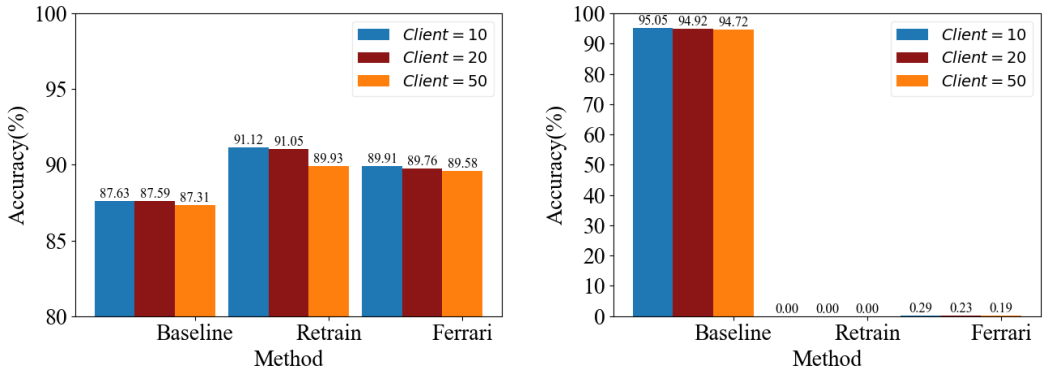
### A.5 Non-IID Analysis

This section presents an analysis of the impact of Non-IID data on the performance of the proposed Ferrari framework compared to the Baseline and Retrain methods on the CIFAR-10 dataset. We focus on the accuracy of the retain client dataset ($\mathcal{D}_r$) and the unlearn client dataset ($\mathcal{D}_u$) in backdoor feature unlearning, as illustrated in Fig.16. To measure the extent of Non-IID, we used the $Dir(\gamma)$ distribution, where smaller values of $\gamma$ indicate more heterogeneous data.

The results show that the Ferrari framework significantly improves feature unlearning performance, with a drop of approximately 0.2% in $\mathcal{D}_u$ when $\gamma = 1$ compared to the IID scenario. Furthermore, the Ferrari framework maintains successfully the accuracy of $\mathcal{D}_r$ with only a slight decrease of about 2% compared to the Retrain method within the Non-IID scenario.

### A.6 Client Numbers Analysis

This section analyzes the impact of a large-scale FL environment, characterized by a large number of clients, on the performance of the proposed Ferrari framework compared to the Baseline and Retrain methods on the CIFAR-10 dataset. We focus on the accuracy of the retained client dataset ($\mathcal{D}_r$) and the unlearned client dataset ($\mathcal{D}_u$) in backdoor feature unlearning, as illustrated in Fig.17. The results indicate that the unlearning performance of our proposed Ferrari framework remains




(a) $\mathcal{D}_r$ (b) $\mathcal{D}_u$

Figure 17: Scability analysis of client numbers on the CIFAR-10 dataset on the accuracy of retain client dataset $D_r$ and unlearn client dataset $D_u$

consistent, with no significant changes in the accuracy of both $\mathcal{D}_r$ and $\mathcal{D}_u$ as the number of clients increases. This finding further demonstrates the effectiveness of the Ferrari framework in large-scale FL environments.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.