Zero-shot Federated Unlearning via Transforming from Data-Dependent to Personalized Model-Centric

Wenhan Wu[1], **Huanghuang Liang**[1], **Jingling Yuan**[2], **Jiawei Jiang**[1], **Kanye Ye Wang**[3], **Chuang Hu**[1,3*], **Xiaobo Zhou**[3] and **Dazhao Cheng**[1*]
[1]School of Computer Science, Wuhan University, Wuhan, China
[2]Hubei Key Laboratory of Transportation Internet of Things, School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China
[3]State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau SAR
{wenhanwu, hhliang, jiawei.jiang, handc, dcheng}@whu.edu.cn, yjl@whut.edu.cn,
{wangye, waynexzhou}@um.edu.mo

Abstract

Federated Unlearning (FU) addresses the "right to be forgotten" in federated learning by removing specific client data's contribution without retraining from scratch. Existing FUs are data-dependent, which make the assumption that systems can access original training data or stored historical parameter updates during unlearning. However, the assumption cannot always hold in practice, as users usually request the deletion of client data and historical parameter updates due to privacy concerns or storage limitations. Therefore, it is crucial to develop a zero-shot FU method without such data access. The key challenge is how to distinguish and remove the impact of target clients without data-level information. Motivated by the idea that if we can learn client-specific personalized information from the model instead of data, FU can be model-centric and data-free, we present the first zero-shot FU framework ZeroFU. By embedding client contributions into the model during learning via condition computation, ZeroFU enables the model to possess personalized features for unlearning. The unlearning is achieved using a proposed GAN-based distillation framework that obfuscates the personalized feature of the target client. Evaluations demonstrate its effectiveness in unlearning under non-IID settings.

1 Introduction

Privacy regulations such as GDPR [Voigt and Von dem Bussche, 2017] and CCPA [Illman and Temple, 2019] grant clients in federated learning (FL) [McMahan *et al.*, 2017] the right to withdraw their data contributions. While the most straightforward solution is to retrain the model from scratch on the retained clients, it is costly and computationally intensive. *Federated Unlearning* (FU) [Wu *et al.*, 2022; Liu *et al.*, 2021] has thus emerged as a critical solution, aiming to more efficiently "forget" the impact of specific data by fine-tuning the trained model or accelerating retraining. According to the target of unlearning, current FUs focus on classes, clients, or samples, aiming to eliminate the impact of data from specific classes, clients, or samples, respectively. FU in this paper refers to federated client-level unlearning.

Existing FUs are *data-dependent*, which make an assumption that systems can re-access training data or model updates while unlearning. They require clients to have access to client-local data for model fine-tuning [Liu *et al.*, 2022; Wang *et al.*, 2022] or reuse stored historical parameter updates during training to accelerate retraining [Zhang *et al.*, 2023c; Liu *et al.*, 2021; Lin *et al.*, 2024]. However, such an assumption cannot always hold in practice. First, for the client requesting unlearning, any access to its data during the unlearning phase should be strictly prohibited, e.g. if a user of a shopping app discovers that its private data has been used to train an FL model, they are likely to immediately revoke access and request the deletion of the data's impact. Second, for retained clients who do not request unlearning, their training data or historical updates are often inaccessible after training, e.g. in the field of medical records, where privacy regulations like HIPAA [Cohen and Mello, 2018] and user consent withdrawal require that such records be permanently deleted after training. Additionally, storing historical updates on the server or client increases storage overhead and poses privacy risks, making it infeasible for large-scale federated deployment. Such situations necessitate **zero-shot federated unlearning**, i.e. *forgetting the contributions of a specific client without visiting the original data or stored updates.*

The core challenge of zero-shot FU lies in how to distinguish the impact of the target client from non-target clients on the model and remove the impact. Existing data-dependent FUs typically achieve contribution differentiation through client data distributions or gradient update differences during training, but such information is unavailable in zero-shot scenarios with no client-specific knowledge available. One potential idea is to be model-centric, i.e. embedding the client-specific knowledge into the model. The key is to embed personalized client features during training to enable model-based unlearning rather than relying on the data itself, which means transforming the FU from data-dependent to a *person-*

*Corresponding Authors.

alized model-centric approach. The personalized features of the model can be used to distinguish the contributions of target clients for effective unlearning.

Motivated by the above ideas, we propose *ZeroFU*, a novel FU framework designed for zero-shot scenarios which is personalized model-centric. ZeroFU innovatively addresses the absence of data or updates after training based on the personalized client information embedded during training. Specifically, during the FL phase, we design a conditional computation mechanism to dynamically identify and utilize the unique contributions of each client within the model. Then during the FU phase, we propose a GAN-based distillation framework. The adversarial structure of the GAN is responsible for generating pseudo data for model distillation. The goal of distillation is to eliminate the influence of the target client while preserving the overall model performance. This approach ensures the contributions of target clients are effectively removed while minimizing catastrophic forgetting [Bourtoule *et al.*, 2021; Liu *et al.*, 2024] of other clients' performance. ZeroFU remains effective and reliable in zero-shot scenarios where existing methods fail to work. Extensive experimental results demonstrate that ZeroFU performs well on both target and retained clients. The main contributions are as follows:

- We propose a novel zero-shot federated unlearning framework, designed to handle scenarios where neither training data nor historical updates are accessible. This aligns closely with real-world data deletion requests.
- We propose utilizing personalized client information embedded in the model during training to effectively achieve unlearning by extracting and obfuscating client-specific features, thereby transitioning FU from data-dependent to personalized model-centric.
- We deploy and evaluate ZeroFU on real-world datasets, achieving an accuracy improvement of up to 26.2% compared to existing zero-shot machine unlearning methods extended to federated scenarios.

2 Related Work

Federated Unlearning. Federated unlearning (FU) extends machine unlearning (MU) to federated scenarios, aiming to remove specific information from trained models for privacy compliance. In addition to the class-level and sample-level forgetting dimensions in MU, FU introduces client-level as an additional dimension. Precise MU methods like SISA [Bourtoule *et al.*, 2021] retrain data partitions, while approximate methods [Lee and Woo, 2023; Tarun *et al.*, 2023] leverage statistical properties. Existing FU methods often require historical parameters or original data. For instance, class-level methods [Wang *et al.*, 2022] prune sensitive channels for specific classes, while client-level methods like FedEraser [Liu *et al.*, 2021] and others [Wu *et al.*, 2022; Liu *et al.*, 2022; Su and Li, 2023] use historical updates, knowledge distillation, or retraining techniques to accelerate unlearning. These methods depend on client access to data post-training, which is often infeasible in practice. Despite these limitations, client unlearning in zero-shot scenarios remains unexplored.

In client-level FU, given a training dataset $D = \{D_i\}_{i=1}^K$ from clients $C_1, C_2, \ldots, C_K$, let C_r represent the retained client with data D_r, and C_f the forgotten (or target) client with data D_f. A *retrained model*, trained from scratch on $D \setminus D_f$, is typically used as a benchmark for unlearning performance [Bourtoule *et al.*, 2021; Liu *et al.*, 2021]. The FL process, denoted as $\mathcal{FL} : D \to \omega$, maps the client data D to the global model parameters ω. The FU process is defined as $\mathcal{FU} : \mathcal{FL}(D) \otimes D \otimes D_f \to \omega'$, where the goal is to produce an unlearned model ω' similar to the retrained model:

$$\Phi[\mathcal{FL}(D \setminus D_f)] = \Phi[\mathcal{FU}(\mathcal{FL}(D), D, D_f)], \tag{1}$$

where $\Phi[\cdot]$ denotes the probability distribution of the output.

Zero-shot Federated Unlearning. Existing FU methods require access to D_r and D_f, which is often impractical due to privacy constraints. In contrast, a zero-shot client unlearning approach ($\mathcal{FU}'$) eliminates dependence on D_r and D_f, relying instead on model queries as follows:

$$\Phi[\mathcal{FL}(D \setminus D_f)] = \Phi[\mathcal{FU}'(\mathcal{M}_r(D), \mathcal{M}_f(D))], \tag{2}$$

where $\mathcal{FU}' : \mathcal{FL}(D) \otimes \mathcal{M}_r \otimes \mathcal{M}_f \to \omega'$ queries the personalized models $\mathcal{M}_r$ and $\mathcal{M}_f$ of the retained and forgotten clients, respectively. This method only requires access to the model parameters ω, which are typically available.

Model Personalization. Existing FL methods like FedAvg [McMahan *et al.*, 2017] train a single global model in a privacy-preserving manner. However, data heterogeneity in real-world scenarios limits the performance of such global aggregation [Wang *et al.*, 2020; Xu *et al.*, 2024]. Model personalization addresses this issue by tailoring the global model for each client. Approaches such as GPFL [Zhang *et al.*, 2023a] and FedCP [Zhang *et al.*, 2023b] improve local accuracy by separating personalized and global information, while others like DPMN [Ma *et al.*, 2022], Peaches [Yan *et al.*, 2024], and TopFL [Chen *et al.*, 2024] utilize personalized model topologies. In this paper, we find that model personalization not only enhances model performance in non-IID scenarios but also facilitates client FU in zero-shot scenarios.

3 Design of ZeroFU

3.1 Overall Architecture

The overall process involves first collaboratively training the FL model, and then during unlearning, the forgotten client sends an unlearning request to the retained clients, followed by executing unlearning to remove its specific contributions.

ZeroFU Learning Framework. As shown in Figure. 1, the components of learning include a Feature Extraction module (θ), a Model Head (φ), a Class Embedding Generator ($eGen$) and a Condition Module (CM). ZeroFU employs a personalized approach that extracts both global and client-specific features. Following the principles of FedRep [Collins *et al.*, 2021], the backbone model, such as ResNet [He *et al.*, 2016], is divided into θ and φ during training. The φ corresponds to the last fully connected (FC) layer of the backbone model, while the θ comprises the remaining layers. Global Embedding gEm and Personal Embedding pEm_i are calculated based on the class embedding (cEm) from $eGen$ and the label distribution (LD_i) of client C_i using a *look up* method. The gEm and pEm_i are then fed into CM. The

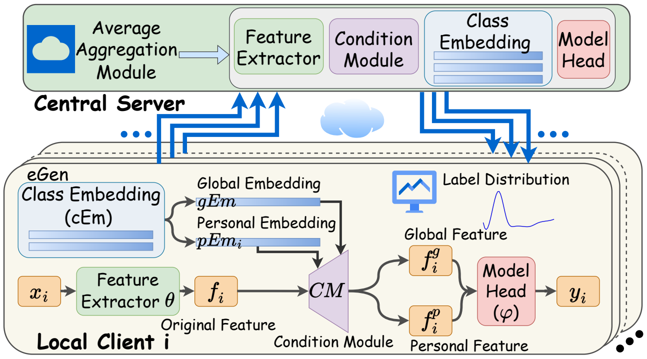


Figure 1: ZeroFU Learning Framework with Condition Module.

CM processes the extracting features $\boldsymbol{f_i}$ from θ and generating both global features $\boldsymbol{f_i^g}$ and client-specific features $\boldsymbol{f_i^p}$, which are then combined to produce the final classification result through φ. θ, φ, CM, and $eGen$ share parameters among clients and are trained in an end-to-end manner. The trainable parameters include $\omega_i = \{\omega^\theta, \omega^\varphi, \omega^{CM}, \omega^{eGen}\}$.

ZeroFU Zero-shot Unlearning Framework. As shown in Figure. 2, the components of unlearning include the personalized model on retained client C_r and forgotten client C_f, a student model and a Generator G. ZeroFU uses Knowledge Distillation (KD) and Generative Adversarial Networks (GAN) to adjust model parameters and obscure the personalized information of C_f. The unlearning process is conducted on the client side for privacy reasons, with the teacher model being the model of C_r. The student model has the same model structure as the model on C_r. We use G to create data points aimed at maximizing the difference $FLoss$ between personalized features $\boldsymbol{f_s^p}$ on student and $\boldsymbol{f_f^p}$ on C_f, while the client local distillation minimizes $FLoss$ to achieve the forgetting of personalized feature information. Intuitively, the personalized information of the client C_f are overshadowed by that of the retained client C_r. Through knowledge distillation, zero-shot unlearning is realized. Knowledge transfer is achieved by minimizing the divergence $KLoss$ and distance measures $Aloss$ between the teacher and student model. CM and $eGen$ remain unchanged to maintain class consistency.

3.2 Learning Personalized Models on ZeroFU

ZeroFU's learning phase aims to train personalized models for each client, extracting client-specific features for unlearning. It involves collaboration between θ, φ, $eGen$, and CM.

Feature Extractor. θ extracts features $\boldsymbol{f_i}$ from the input $\boldsymbol{x_i}$ on client C_i. θ is a backbone network module excluding the model head φ, mapping $\theta : \mathbb{R}^{\dim(\boldsymbol{x})} \to \mathbb{R}^{\dim(\boldsymbol{f_i})}$, where typically data dimension $\dim(\boldsymbol{f_i}) \ll \dim(\boldsymbol{x})$. For client i,

$$\forall(\boldsymbol{x_i}, y_i) \in D_i, \boldsymbol{f_i} = \theta(\boldsymbol{x_i}; \omega^\theta). \tag{3}$$

Generating Class Embedding. We design the embedding generator $eGen$. $eGen$ generates global class embedding vectors (cEm) to compute Global Embedding (gEm) and Personal Embedding (pEm_i) as the condition for the condition module CM. For an image classification task with U

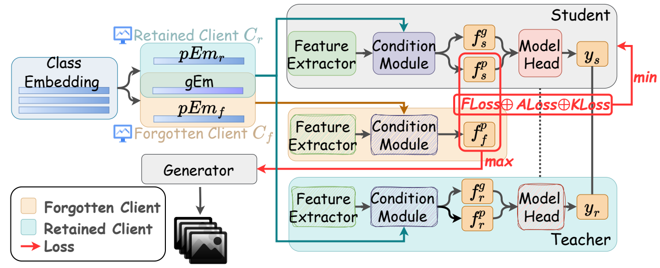


Figure 2: ZeroFU Zero-shot Unlearning Framework.

classes, $eGen$ takes a class index input $cin \in \{0, 1, \ldots, U-1\}$ and outputs the embedding cEm. The output of $eGen$ is:

$$cEm_y = eGen(cin = y; \omega^{eGen}), \tag{4}$$

where $cEm_y \in \mathbb{R}^{\dim(\boldsymbol{f_i})}$ is the embedding vector for class y. The global class embedding matrix is $cEm = [cEm_0, cEm_1, \ldots, cEm_{U-1}]$, where $cEm \in \mathbb{R}^{U\times\dim(\boldsymbol{f_i})}$.

Generating Global and Personal Embeddings. Based on cEm, we generate gEm and pEm_i for each client C_i. The gEm captures the global information of all classes and is shared across all clients. It is calculated as:

$$gEm = \frac{1}{U}\sum_{y=0}^{U-1} cEm_y. \tag{5}$$

Let the label distribution vector of client C_i be $LD_i = [LD_i^0, LD_i^1, \ldots, LD_i^{U-1}]$, where:

$$LD_i^y = \mathbb{E}_{(\boldsymbol{x}_i, y_i)\sim\mathcal{D}_i}\mathbb{I}\{y_i, y\}, \tag{6}$$

where $\mathbb{I}$ is an indicator function. The personalized embedding pEm_i is calculated based on the LD_i as:

$$pEm_i = \sum_{u=0}^{U-1}(cEm_u \cdot LD_i^u). \tag{7}$$

The pEm_i reflects the specific label distribution information of C_i, providing conditions for non-IID data in FL settings.

Generating Personalized and Global Features. Inspired by conditional computation techniques [Guo *et al.*, 2019; Zhang *et al.*, 2023a], the function of CM is to utilize the conditions from gEm and pEm_i to generate personalized feature representations $\boldsymbol{f_i}^p$ and global feature representations $\boldsymbol{f_i}^g$ from the input feature $\boldsymbol{f_i}$, respectively. This enables personalized model adjustments for each client. The CM includes a Conditional Bias module CM_B and a Conditional Weight module CM_W. For the global feature extraction, the output weight vector $\mathbf{W}$ and bias vector $\mathbf{b}$ are:

$$\mathbf{W} = CM_W(gEm; \omega^{CM}), \mathbf{b} = CM_B(gEm; \omega^{CM}). \tag{8}$$

For the personal feature extraction, the output vectors are:

$$\mathbf{W_i} = CM_W(pEm_i; \omega^{CM}), \mathbf{b_i} = CM_B(pEm_i; \omega^{CM}). \tag{9}$$

We compute $\boldsymbol{f_i^g}$ and $\boldsymbol{f_i^p}$ by performing affine transformations [Zhang *et al.*, 2023b; Zhang *et al.*, 2023a] as follows:

$$\begin{aligned} \boldsymbol{f_i^g} &= \mathrm{ReLU}\left(\mathbf{b} + (\mathbf{W} + \mathbf{1}) \odot \boldsymbol{f_i}\right), \\ \boldsymbol{f_i^p} &= \mathrm{ReLU}\left(\mathbf{b}_i + (\mathbf{W}_i + \mathbf{1}) \odot \boldsymbol{f_i}\right), \end{aligned} \tag{10}$$

where $\odot$ denotes Hadamard product, $\mathbf{1}$ is an all-ones matrix.

Model Head Output. We concatenate $\boldsymbol{f_i^p}$ and $\boldsymbol{f_i^g}$, and then the model head φ converts the concatenated feature vector into the final prediction result $\hat{y}_i$:

$$\boldsymbol{\hat{y}_i} = \varphi([\boldsymbol{f_i^p}; \boldsymbol{f_i^g}]; \omega^{\varphi}). \tag{11}$$

Training Loss Function. ZeroFU's loss function combines multiple terms. The primary loss for classification tasks is the cross-entropy (CE) loss [Mannor *et al.*, 2005]:

$$\mathcal{L}_i^{\text{CE}} = CrossEntropyLoss(\boldsymbol{y_i}, \boldsymbol{\hat{y}_i}), \tag{12}$$

where $\boldsymbol{\hat{y}_i}$ is calculated according to Eq. (11). Additionally, inspired by the class embedding loss functions [Pu *et al.*, 2024], we incorporate an extra loss term for cEm to adjust the update direction of $cGen$. We aim to minimize the similarity between the feature vector and its corresponding class embedding while maximizing its distance from other class embeddings, thereby preserving the specificity of different class embeddings. We calculate the cosine similarity between $\boldsymbol{f_i^g}$ and its corresponding class embedding as follows:

$$\mathcal{L}_i^{\text{EM}} = -\log\left(\frac{\exp(cos\_sim(\boldsymbol{f_i^g}, cEm_{y_i}))}{\sum_{u=1}^{U}\exp(cos\_sim(\boldsymbol{f_i^g}, cEm_u))}\right), \tag{13}$$

where the cosine similarity is calculated as the formula $cos\_sim(f, emb) = \frac{f \cdot emb^{\text{T}}}{\|f\| \cdot \|emb\|}$. Combining the loss functions above, the overall loss function for client C_i is:

$$\mathcal{L}_i(\omega_i) = \mathcal{L}_i^{\text{CE}} + \mathcal{L}_i^{\text{EM}} + \lambda_1\|\omega^{cGen}\|_2^2 + \lambda_2\|\omega^{CM}\|_2^2, \tag{14}$$

where we add L2 regularization terms to the $cGen$ and CM, with λ_1, λ_2 as hyperparameters. We minimize the loss function using SGD and aggregate the updates. The entire learning process is outlined in Algorithm 1.

3.3 Zero-shot Unlearning Process on ZeroFU

ZeroFU aims to efficiently remove the influence of a forgotten client C_r on a retained client C_r by forgetting personalized features in a zero-shot setting based on KD and GAN.

Data Generator. We use a generator $G(z, \omega_{\text{G}})$ to generate data points from a random noise vector $z \in \mathbb{R}^{z_{\text{dim}}}$, sampled from $\mathcal{N}(0, 1)$. The generator first maps z to a higher dimensional space using a series of layers, including Linear, View, Batch Normalization, and Upsampling. It then processes the feature maps with convolution, batch normalization, and Leaky RELU activation to generate pseudo-samples for knowledge distillation training, denoted as $\boldsymbol{x} = G(z; \omega_{\text{G}})$. These pseudo-samples are then used as inputs for the teacher model, student model, and the model to be forgotten on C_f.

Zero-shot Knowledge Distillation. ZeroFU uses the retained model $R(\boldsymbol{x}; \omega_r)$ on C_r as the teacher and a randomly initialized model $S(\boldsymbol{x}; \omega_s)$ as the student. The model to be forgotten on C_f is $F(\boldsymbol{x}; \omega_f)$. To maintain training accuracy, CM and $cGen$ parameters are kept unchanged, $\omega_s^{CM} = \omega_r^{CM}$, $\omega_s^{cGen} = \omega_r^{cGen}$. The objective is to maintain performance on C_r while unlearning C_f's information. Inspired by [Chen *et al.*, 2023; Lee and Woo, 2023], we aim to alter the decision space on C_f by mapping its personalized information decision space to the C_r's personalized information decision space, achieving approximate unlearning. Specifically, we aim to make the personalized feature information $\boldsymbol{f_f^p}$ extracted from client C_f close to the $\boldsymbol{f_s^p}$ of the student. We obtain $\boldsymbol{f_s^p}$ and $\boldsymbol{f_f^p}$ from $S(\boldsymbol{x}; \omega_s)$ and $F(\boldsymbol{x}; \omega_f)$ using Eq. (9) and (10), respectively. We have set $pEm_s = pEm_r$, using the personal embedding of the client C_r on the student model. The similarity between $\boldsymbol{f_s^p}$ and $\boldsymbol{f_f^p}$ is as follows:

$$FLoss = 1 - cos\_sim(\boldsymbol{f_s^p}, \boldsymbol{f_f^p}) = 1 - \frac{\boldsymbol{f_s^p} \cdot \boldsymbol{f_f^p}^{\text{T}}}{\|\boldsymbol{f_s^p}\| \cdot \|\boldsymbol{f_f^p}\|}, \tag{15}$$

where a smaller $FLoss$ indicates a higher cosine similarity, achieving "masking" of C_f's personalized information without affecting the decision space of the C_r. Similarly, we can compute $\boldsymbol{f_r^p}$ (the personalized information feature on C_r) by using pEm_r. The $\boldsymbol{f_s^g}$ and $\boldsymbol{f_r^g}$ (global feature information of student and teacher models) can be calculated by replacing pEm with gEm. Let $\hat{y_r}$ and $\hat{y_s}$ be the predictions of the teacher and student models on x respectively. To enable the knowledge transfer from teacher to student, $KLoss$ maximizes the Kullback-Leibler divergences [Kullback and Leibler, 1951] $D_{KL}(R(\boldsymbol{x})\|S(\boldsymbol{x}))$ between the outputs:

$$KLoss = \tau^2 \sum_{i=0}^{U-1} \sigma(\hat{y_r}/\tau)_i \log\left(\frac{\sigma(\hat{y_r}/\tau)_i}{\log\sigma(\hat{y_s}/\tau)_i}\right), \tag{16}$$

where τ is a temperature parameter, σ represents the softmax function. Additionally, inspired by [Micaelli and Storkey, 2019], we incorporate attention loss with L2 normalization:

$$ALoss = \frac{1}{|\mathcal{N}_L|}\sum_{l\in\mathcal{N}_L}\left\|\frac{f(A_l^{(r)})}{\|f(A_l^{(r)})\|_2} - \frac{f(A_l^{(s)})}{\|f(A_l^{(s)})\|_2}\right\|_2, \tag{17}$$

Algorithm 1: Model Personalization Based Training

Input: Client C_i training data D_i; Learning rate η; Hyperparameters λ_1, λ_2; Total communication rounds T; Local training rounds T_l; Ratio of joining clients per round α.
Output: Updated global model parameters ω^T for K clients.

```
Initialize global parameters shared among clients with:
  ω^0 ← {ω^{θ,0}, ω^{φ,0}, ω^{CM,0}, ω^{eGen,0}}.
for each communication round t from 1 to T do
    Randomly select a fraction α of clients to form set O^t.
    Server sends ω^{t-1} to client ∀C_i ∈ O^t, set ω_i^t ← ω^{t-1}.
    for each selected client C_i ∈ O^t in parallel do
        for each local training round t_l from 1 to T_l do
            Extract features f_i from θ using Eq. (3).
            Generate gEm, pEm using Eq. (5),(7).
            Compute f_i^g and f_i^p from CM using Eq. (10).
            Compute prediction ŷ_i from φ using Eq. (11).
            Compute loss function L_i(ω_i) using Eq. (14).
            Update local parameters using gradient
              descent: ω_i^t ← ω_i^t − η∇L_i(ω_i).
        Share updated parameters ω_i^t with the server.
    Aggregate global parameters on the server:
      ω^t ← 1/(Σ_{i∈O^t} |D_i|) Σ_{i∈O^t} |D^i| ω_i^t.
return Updated global model ω^T.
```

Algorithm 2: ZeroFU Zero-shot Unlearning Process

Input: Teacher model on C_r: $R(\boldsymbol{x};\omega_r)$; Forgetting model on C_f: $F(\boldsymbol{x};\omega_f)$; Generator: $G(z;\omega_{\mathrm{G}})$; Hyperparameters β and γ; Unlearning training rounds T_u; KD rounds T_k; Learning rate η.
Output: Updated student model $S(\boldsymbol{x};\omega_s)$.

```
Initialize student model S(x; ω_s) with random weights ω_s.
Set ω_s^{CM} ← ω_r^{CM}, ω_s^{cGen} ← ω_r^{cGen}.
for each unlearning round t from 1 to T_u do
    Sample a batch of random noise z ~ N(0, 1).
    Generate pseudo-samples x ← G(z; ω_G).
    for each KD round k from 1 to T_k do
        Compute personalized features f_r^p, f_s^p, f_f^p.
        Compute global information features f_r^g and f_s^g.
        Compute FLoss using Eq. (15).
        Compute predictions ŷ_r and ŷ_s using Eq. (11).
        Compute KLoss using Eq. (16).
        Compute attention loss ALoss using Eq. (17).
        Compute S(x; ω_s) total loss: F_stu using Eq. (19).
        Update S(x; ω_s) parameters using gradient
          descent: ω_s^{θ,φ} ← ω_s^{θ,φ} − η∇_{ω_s^{θ,φ}} F_stu.
    Update G(z; ω_G) parameters ω_G using gradient ascent:
      ω_G ← ω_G + η∇_{ω_G} FLoss.
return Updated student model S(x; ω_s).
```

where $A_l^{(r)}$ and $A_l^{(s)}$ denote the outputs of the teacher and student models at layer l, respectively, and $\mathcal{N}_L$ represents the subset of layers used for calculating the attention loss, specifically the activation layers of θ. A_l outputs contain n_l channels. The function $f(A_l) = \frac{1}{n_l}\sum_c a_{l,c}^2$, where $a_{l,c}$ denotes the c-th channel of the activation block A_l.

GAN Training Process. The generator G is optimized to maximize the difference between the personalized information features of the model $F(x;\omega_f)$ on the C_f and the student model $R(x;\omega_r)$. Here, the personalized feature information acts as the discriminator in the GAN. The goal of G is:

$$\arg\max_{\omega^G} FLoss. \tag{18}$$

During knowledge distillation, the student model updates its weights with an adversarial goal of forgetting specific personalized information on C_f, i.e., minimizing $FLoss$. With the hyperparameters β and γ, the total loss for the student is:

$$\arg\min_{\omega_s^\theta,\omega_s^\varphi} F_{stu}, \text{where } F_{stu} = FLoss + \beta KLoss + \gamma ALoss, \tag{19}$$

The entire unlearning process is outlined in Algorithm 2.

4 Evaluation

4.1 Methodology

Testbed. We deployed ZeroFU on NVIDIA A100 40GB Tensor Core GPUs using PyTorch 2.3.1 and Python 3.8.

Datasets and Backbone Models. Datasets include: MNIST [LeCun *et al.*, 1998], SVHN [Netzer *et al.*, 2011], Fashion-MNIST [Xiao *et al.*, 2017], and CIFAR10 [Krizhevsky and Hinton, 2009]. The backbone model includes two conv layers (32 and 64 5×5 filters with ReLU and 2×2 max-pooling) followed by a linear output layer.

Baselines. Given the lack of zero-shot unlearning in federated environments, we adapted zero-shot methods from MU for federated unlearning and compared them under ZeroFU's learning configuration in a non-IID setting: **i). FedMM:** Employs the *Error Minimization-Maximization Noise* method [Tarun *et al.*, 2023; Chundawat *et al.*, 2023b], learning noise matrices to minimize $FLoss$ for retained clients' personalized features while maximizing it for forgotten clients' features. **ii). FedGKT:** Extends *Gated Knowledge Transfer* [Chundawat *et al.*, 2023b] to FL. It filters out forgotten client information via a band-pass filter between personalized features $\boldsymbol{f_r^p}$ and $\boldsymbol{f_f^p}$. **iii). FedBadT**: Utilizes both competent and incompetent teacher models [Chundawat *et al.*, 2023a], retaining the zero-shot data generator. The retained client model acts as the competent teacher, while the forgotten client outputs random noise as the incompetent teacher. We also compared with non-zero-shot FUs in §4.3.

Evaluation Metrics. The evaluation metrics are: **i). Accuracy**: The accuracy of D_f and D_r for C_f and C_r should be closer to that of the retrained model, as the expected behavior of the unlearned model should resemble the retrained model. **ii). Weight Distance**: We measure model similarity using L2 Norm Distance [Thudi *et al.*, 2022], with smaller differences indicating higher similarity. **iii). Membership Inference Attack (MIA)**: MIA, framed as a binary classification task using a logistic regressor [Yan *et al.*, 2022], determines if a data point in D_f was used in training [Chen *et al.*, 2023; Liu *et al.*, 2021]. The threat model, trained on $\boldsymbol{f_i^p}$ extracted by unlearned model on D_r with the labels indicating whether this data was used during training, aims to match retrained model performance, indicating reduced attack capability. Metrics include *attack precision* and *recall*.

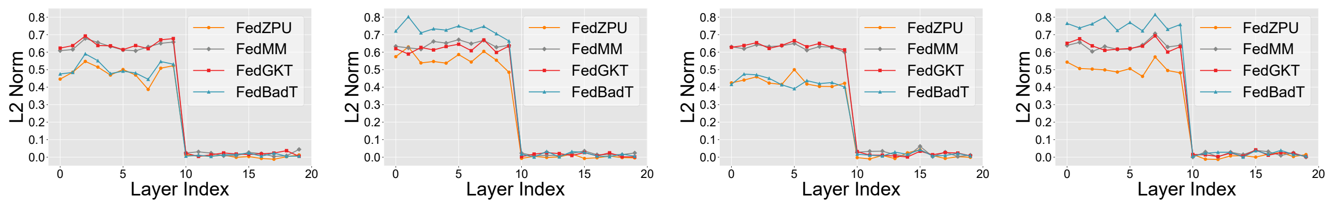
Settings and Hyperparameters. We used 10 clients with $\alpha = 1$ per round. To simulate the FL environment, we applied label shift heterogeneity under two settings: **iii). Pathological label skew**: Data with different labels and sizes were sampled for each client [McMahan *et al.*, 2017]. **ii). Practical label skew**: A Dirichlet distribution ($Dir(\zeta)$, with the concentration parameter $\zeta = 0.01/0.1$) was used [Lin *et al.*, 2020]. The learning rate was $\eta = 0.005$ with $T_l = 3$, and regularization parameters $\lambda_1 = \lambda_2 = 0.1$. During FU, $\tau = 2$, $T_k = 9$, and the loss hyperparameters were $\beta = 5.0$, $\gamma = 2.0$.

4.2 Overall Performance

Performance on Retained and Forgotten Data. Table 1 compares ZeroFU's accuracy with baseline methods on retained data D_r and forgotten data D_f. Under $\zeta = 0.01$ (highly imbalanced data, simulating class unlearning [Zhang *et al.*, 2023c]), ZeroFU achieves an average difference of 0.87% on D_r and 0.44% on D_f, outperforming FedMM (11.34%, 2.46%), FedGKT (8.43%, 2.44%), and FedBadT (2.59%, 47.15%). For $\zeta = 0.10$ (overlapping label distributions), ZeroFU maintains differences of 3.21% on D_r and 0.44% on D_f, while baselines exceed 20%. FedBadT struggles on FMNIST and CIFAR10 due to unbalanced KL divergence between two teachers. FedGKT suffers catastrophic forgetting under $\zeta = 0.10$ due to its gating mechanism blocking same-label propagation, while FedMM's noise method re-

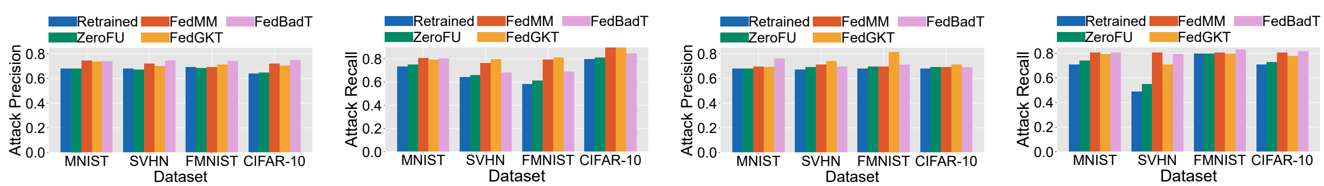
DataSet	ζ	C_r	C_f	Origin		Retrained		ZeroFU		FedMM		FedGKT		FedBadT	
				D_r	D_f	D_r	D_f	D_r	D_f	D_r	D_f	D_r	D_f	D_r	D_f
MNIST	0.01	0	1	96.40	99.22	96.30	0.24	92.63	0.00	73.11	0.00	71.14	0.00	88.75	0.00
		8	9	98.54	99.91	98.81	0.01	97.05	0.01	89.11	0.00	89.11	0.00	89.11	1.13
	0.1	4	5	99.10	99.26	99.03	57.18	96.03	38.95	20.98	0.00	60.70	0.12	82.66	27.53
		3	6	96.02	97.29	94.41	83.79	91.04	83.54	44.89	65.19	54.60	0.00	85.00	73.23
SVHN	0.01	0	1	96.02	95.13	96.02	0.00	96.02	0.00	96.02	0.00	96.02	0.00	96.02	4.96
		8	9	95.99	99.99	95.84	0.00	95.99	0.00	95.99	0.00	93.44	0.00	95.99	8.95
	0.1	4	5	98.13	70.87	98.13	60.26	98.13	46.60	98.13	36.98	98.13	39.98	98.13	38.10
		3	6	88.23	59.93	86.08	54.60	88.26	45.41	88.16	0.00	88.16	0.00	88.16	8.40
FMNIST	0.01	0	1	99.45	99.06	99.45	6.27	99.45	8.69	100.00	12.70	71.95	0.00	100.00	96.28
		8	9	99.98	99.44	99.50	12.96	99.98	12.13	99.99	0.00	99.98	0.00	99.98	84.92
	0.1	4	5	83.12	85.91	88.13	13.35	79.78	16.11	63.12	16.11	59.41	16.11	59.40	83.48
		3	6	76.74	98.11	88.41	13.88	84.81	17.31	49.61	0.00	21.04	0.00	73.84	95.41
CIFAR10	0.01	0	1	98.78	100.00	98.78	0.00	98.78	0.00	98.78	0.00	98.78	0.00	98.78	100.00
		8	9	80.36	99.96	77.88	0.00	78.76	0.00	21.28	0.00	75.68	0.00	80.13	99.96
	0.1	4	5	86.08	80.27	79.77	28.22	78.10	29.01	71.01	7.07	48.70	2.30	71.11	12.11
		3	6	87.19	91.70	80.15	3.76	76.67	2.01	47.48	0.00	41.20	9.58	65.30	92.21

Table 1: Forgetting Accuracy Results Comparison under Different Datasets with the Red Numbers the Optimal Results.



(a) MNIST ($\zeta = 0.01$) (b) MNIST ($\zeta = 0.10$) (c) SVHN ($\zeta = 0.01$) (d) SVHN ($\zeta = 0.10$)

Figure 3: Deviation of Parameters between Unlearned Model and Retrained Model under Different Methods.



(a) Attack Precision ($\zeta = 0.01$) (b) Attack Recall ($\zeta = 0.01$) (c) Attack Precision ($\zeta = 0.10$) (d) Attack Recall ($\zeta = 0.10$)

Figure 4: Performance of Membership Inference Attacks (MIAs).

duces accuracy and amplifies forgetting. ZeroFU effectively balances forgetting and retention by disrupting D_f's decision boundary without impacting other clients' boundaries.

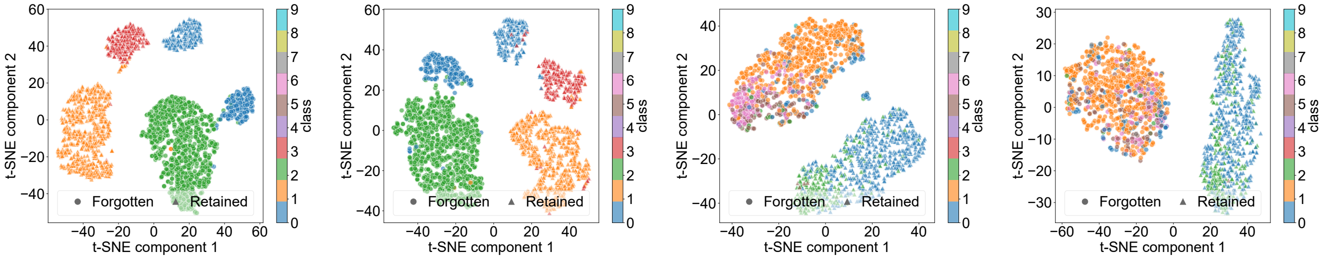
Weight Deviation of Unlearned Models. The comparisons between the weights of unlearned and retrained models are shown in Figure. 3. The weights of the unlearned models created using ZeroFU are generally closer to the retrained models, demonstrating better performance. The parameter deviation decreases when approaching the model head.

Privacy Leakage of Forgotten Data. We conducted MIAs on D_f in non-IID scenarios. As shown in Figure 4, ZeroFU achieves MIA accuracy and recall closer to retrained models, demonstrating better privacy protection. Unlearned models generally leak more privacy due to reliance on original models than retrained models. At $\zeta = 0.01$, FedMM, FedGKT, and FedBadT averaged 4.15%, 3.39%, and 6.60% higher accuracy, and 10.65%, 11.58%, and 4.55% higher recall than ZeroFU. At $\zeta = 0.10$, their attack accuracy averaged 0.90%, 5.04%, and 2.54% higher, with recall 9.89%, 6.63%, and 10.77% higher. This is because ZeroFU more effectively conceals the impact of the forgotten information.

Visualization. We conducted visualization on MNIST and CIFAR10 with $\zeta = 0.10$, using t-SNE [Van der Maaten and Hinton, 2008] to map personalized feature $\boldsymbol{f}_i^{\boldsymbol{p}}$ of retrained and unlearned models. As shown in Figure 5, unlearned models closely resemble retrained models in feature space distribution, appearing isomorphic but not isometric. Furthermore, we observed that the ZeroFU model has the capability to map the same label data from different clients to distinct feature spaces. This prevents catastrophic forgetting of the same label in other clients when a specific label from one client is forgotten. The differentiation in feature spaces across clients also highlights the capacity for learning in non-IID scenarios.

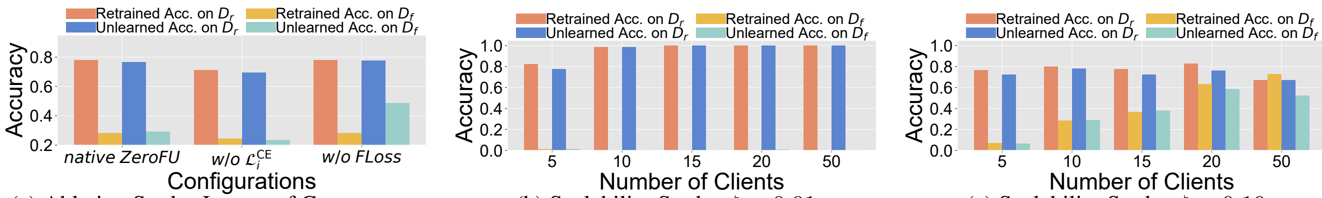
4.3 Comparison with the SOTA FU Methods

We summarize FU method attributes. FedEraser [Liu *et al.*, 2021] uses historical gradients to accelerate retraining, FedRecovery [Zhang *et al.*, 2023c] relies on historical parameters for differential privacy forgetting, and Knot [Su and



(a) MNIST Retrained (b) MNIST Unlearned (c) CIFAR10 Retrained (d) CIFAR10 Unlearned

Figure 5: t-SNE Visualizations of Personalized Features for Unlearned and Retained Clients.



(a) Ablation Study: Impact of Components (b) Scalability Study: $\zeta = 0.01$ (c) Scalability Study: $\zeta = 0.10$

Figure 6: Ablation and Scalability Study: Impact of Components of ZeroFU and Number of Clients.

Dataset	Model	FedEraser D_r	FedEraser D_f	FedRecovery D_r	FedRecovery D_f	Knot D_r	Knot D_f	ZeroFU D_r	ZeroFU D_f
MNIST	retrain	99.01	98.44	99.01	98.44	99.15	98.48	97.78	96.53
	unlearn	96.23	95.84	92.08	91.89	97.66	97.12	94.05	94.77
SVHN	retrain	94.07	89.07	94.07	89.07	97.89	89.36	93.81	89.01
	unlearn	87.55	87.50	81.24	78.90	94.97	93.18	90.45	89.03
FMNIST	retrain	93.79	91.09	93.79	91.09	93.97	90.99	92.24	92.75
	unlearn	90.36	90.09	86.42	83.78	86.27	85.90	91.46	90.88
CIFAR10	retrain	88.73	68.23	88.73	68.23	90.65	87.75	86.70	85.38
	unlearn	85.08	63.57	75.34	63.21	85.33	85.28	85.43	83.78

(a) IID Scenario

Dataset	Model	FedEraser D_r	FedEraser D_f	FedRecovery D_r	FedRecovery D_f	Knot D_r	Knot D_f	ZeroFU D_r	ZeroFU D_f
MNIST	retrain	98.72	87.79	98.72	87.79	97.37	88.72	94.41	83.79
	unlearn	77.16	65.89	90.78	82.36	94.34	85.46	91.04	83.54
SVHN	retrain	84.26	59.26	84.26	59.26	81.95	79.05	86.08	54.60
	unlearn	57.81	42.94	70.12	40.25	80.31	77.63	88.26	45.41
FMNIST	retrain	58.46	22.49	58.46	22.49	58.22	12.55	88.41	13.88
	unlearn	42.32	15.70	49.56	10.98	48.09	38.21	84.81	17.31
CIFAR10	retrain	52.79	39.58	52.79	39.58	71.31	33.89	79.77	28.22
	unlearn	44.10	34.54	40.78	32.08	53.41	33.91	78.10	29.01

(b) non-IID Scenario ($\zeta = 0.10$)

Table 2: Performance Comparison of ZeroFU with State-of-the-Art Client Unlearning Methods: Considering It Cannot Access the Training Dataset, ZeroFU Performs Quite Well. The Red Numbers Represent the Optimal Values, While the Blue Ones Indicate the Second-best.

Li, 2023] employs sub-cluster retraining in asynchronous FL. ZeroFU is unique in not requiring access to training data or historical updates. Performance comparisons (Table 2) show ZeroFU achieves similar unlearning results under IID scenarios. For MNIST, ZeroFU's deviation from the retrained model on D_r and D_f is 3.73% and 2.76%, respectively, compared to Knot's 1.39% and 1.36%. In non-IID scenarios, ZeroFU delivers higher training accuracy, outperforming others by 30% on FMNIST. This is because personalization is conducive to the adaptation of non-IID data. ZeroFU's unlearning effectiveness surpasses FedEraser and FedRecovery and is comparable to Knot. When training data and historical updates are unavailable, ZeroFU remains effective, while others cannot work due to the lack of client-specific information.

4.4 Ablation and Scalability Study

We first conducted ablation experiments using two variants: (a) ***w/o*** $\mathcal{L}_i^{\mathrm{CE}}$, removing the class embedding loss, and (b) ***w/o*** $FLoss$. Results on CIFAR10 are shown in Figure 6a: (a) Removing $\mathcal{L}_i^{\mathrm{CE}}$ reduced learning accuracy by 6.89% due to the loss of global guidance, though unlearning remained effective due to feature remapping by the conditional model CM. (b) Without $FLoss$, unlearning effectiveness dropped, with D_f accuracy 20.32% higher than the retrained model, showing $FLoss$ is critical for disrupting D_f's decision boundaries. For scalability, we tested ZeroFU with 5/10/15/20/50 clients in non-IID settings (Figure 6). In the highly imbalanced $\zeta = 0.01$ scenario, learning and unlearning performed well. Under $\zeta = 0.10$, learning accuracy was stable for 5-20 clients but dropped 12.54% with 50 clients compared to 10. For unlearning, D_r accuracy remained strong as client numbers increased, but D_f accuracy showed a 19.16% gap from the retrained model with 50 clients. This is because as the number of clients increases, overlapping data distributions may retain contributions from clients similar to the target.

5 Conclusion

This paper addressed the pressing need for zero-shot FU, a critical capability in FL systems to comply with privacy regulations and client data withdrawal requests. Unlike existing data-dependent FUs that rely on access to training data or stored updates, ZeroFU pioneers a personalized model-centric approach, embedding client-specific contributions directly into the model to empower unlearning. This study provides a new perspective to advance FU research.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2023YFE0205700), National Natural Science Foundation of China (62341410,62302348), the Macao Science and Technology Development Fund (File no. 0078/2023/AMJ), UM projects (SKL-IOTSC-2024-2026, CPG2025-00029-IOTSC), General Program of Hubei Provincial Natural Science Foundation of China (2023AFB831), the Fundamental Research Funds for the Central Universities (2042025kf0040), Wuhan Knowledge Innovation Special Project (2023020201020250), Shanghai University of Finance and Economics Key Laboratory of Interdisciplinary Research of Computation and Economics, Ministry of Education and Hubei Key Laboratory of Transportation Internet of Things.

References

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

[Chen *et al.*, 2023] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7766–7775, 2023.

[Chen *et al.*, 2024] Suo Chen, Yang Xu, Hongli Xu, Zhenguo Ma, and Zhiyuan Wang. Enhancing decentralized and personalized federated learning with topology construction. *IEEE Transactions on Mobile Computing*, 2024.

[Chundawat *et al.*, 2023a] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 7210–7217, 2023.

[Chundawat *et al.*, 2023b] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 2023.

[Cohen and Mello, 2018] I Glenn Cohen and Michelle M Mello. Hipaa and protecting health information in the 21st century. *Jama*, 320(3):231–232, 2018.

[Collins *et al.*, 2021] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning (ICML)*, pages 2089–2099, 2021.

[Guo *et al.*, 2019] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4805–4814, 2019.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[Illman and Temple, 2019] Erin Illman and Paul Temple. California consumer privacy act. *The Business Lawyer*, 75(1):1637–1646, 2019.

[Krizhevsky and Hinton, 2009] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

[Kullback and Leibler, 1951] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lee and Woo, 2023] Sangyong Lee and Simon S Woo. Undo: Effective and accurate unlearning method for deep neural networks. In *Proc. of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 4043–4047, 2023.

[Lin *et al.*, 2020] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

[Lin *et al.*, 2024] Yijing Lin, Zhipeng Gao, Hongyang Du, Dusit Niyato, Jiawen Kang, and Xiaoyuan Liu. Incentive and dynamic client selection for federated unlearning. In *Proc. of the ACM on Web Conference 2024 (WWW)*, pages 2936–2944, 2024.

[Liu *et al.*, 2021] Gaoyang Liu, Xiaoqiang Ma, Yang Yang, Chen Wang, and Jiangchuan Liu. Federaser: Enabling efficient client-level data removal from federated learning models. In *IEEE/ACM 29th International Symposium on Quality of Service (IWQoS)*, pages 1–10, 2021.

[Liu *et al.*, 2022] Yi Liu, Lei Xu, Xingliang Yuan, Cong Wang, and Bo Li. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 1749–1758, 2022.

[Liu *et al.*, 2024] Ziyao Liu, Yu Jiang, Jiyuan Shen, Minyi Peng, Kwok-Yan Lam, Xingliang Yuan, and Xiaoning Liu. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1):1–38, 2024.

[Ma *et al.*, 2022] Zhenguo Ma, Yang Xu, Hongli Xu, Jianchun Liu, and Yinxing Xue. Like attracts like: Personalized federated learning in decentralized edge computing. *IEEE Transactions on Mobile Computing*, 23(2):1080–1096, 2022.

[Mannor *et al.*, 2005] Shie Mannor, Dori Peleg, and Reuven Rubinstein. The cross entropy method for classification. In

International Conference on Machine Learning (ICML), pages 561–568, 2005.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

[Micaelli and Storkey, 2019] Paul Micaelli and Amos J Storkey. Zero-shot knowledge transfer via adversarial belief matching. *Advances in Neural Information Processing Systems*, 32, 2019.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning (NIPS Workshop)*, page 7, 2011.

[Pu *et al.*, 2024] Nan Pu, Wenjing Li, Xingyuan Ji, Yalan Qin, Nicu Sebe, and Zhun Zhong. Federated generalized category discovery. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28741–28750, 2024.

[Su and Li, 2023] Ningxin Su and Baochun Li. Asynchronous federated unlearning. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 1–10, 2023.

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

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

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

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

[Wang *et al.*, 2020] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE conference on computer communications (INFOCOM)*, pages 1698–1707, 2020.

[Wang *et al.*, 2022] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proc. of the ACM Web Conference 2022 (WWW)*, pages 622–632, 2022.

[Wu *et al.*, 2022] Leijie Wu, Song Guo, Junxiao Wang, Zicong Hong, Jie Zhang, and Yaohong Ding. Federated unlearning: Guarantee the right of clients to forget. *IEEE Network*, 36(5):129–135, 2022.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Xu *et al.*, 2024] Yang Xu, Yunming Liao, Lun Wang, Hongli Xu, Zhida Jiang, and Wuyang Zhang. Overcoming noisy labels and non-iid data in edge federated learning. *IEEE Transactions on Mobile Computing*, 2024.

[Yan *et al.*, 2022] Hongyang Yan, Shuhao Li, Yajie Wang, Yaoyuan Zhang, Kashif Sharif, Haibo Hu, and Yuanzhang Li. Membership inference attacks against deep learning models via logits distribution. *IEEE Transactions on Dependable and Secure Computing*, 20(5):3799–3808, 2022.

[Yan *et al.*, 2024] Jiaming Yan, Jianchun Liu, Hongli Xu, Zhiyuan Wang, and Chunming Qiao. Peaches: Personalized federated learning with neural architecture search in edge computing. *IEEE Transactions on Mobile Computing*, 2024.

[Zhang *et al.*, 2023a] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, Jian Cao, and Haibing Guan. Gpfl: Simultaneously learning global and personalized feature information for personalized federated learning. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5041–5051, 2023.

[Zhang *et al.*, 2023b] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedcp: Separating feature information for personalized federated learning via conditional policy. In *Proc. of ACM Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 3249–3261, 2023.

[Zhang *et al.*, 2023c] Lefeng Zhang, Tianqing Zhu, Haibin Zhang, Ping Xiong, and Wanlei Zhou. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security*, 2023.