Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Federated mutual learning: a collaborative machine learning method for heterogeneous data, models, and objectives*

Tao SHEN[1], Jie ZHANG[2], Xinkang JIA[2], Fengda ZHANG[1],
Zheqi LV[1], Kun KUANG[1], Chao WU[†‡3], Fei WU[†‡1]

[1]*College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*
[2]*School of Software Technology, Zhejiang University, Hangzhou 310027, China*
[3]*School of Public Affairs, Zhejiang University, Hangzhou 310027, China*
[†]E-mail: chao.wu@zju.edu.cn; wufei@zju.edu.cn

Received Feb. 20, 2023; Revision accepted Apr. 7, 2023; Crosschecked May 5, 2023; Published online Aug. 5, 2023

Abstract: Federated learning (FL) is a novel technique in deep learning that enables clients to collaboratively train a shared model while retaining their decentralized data. However, researchers working on FL face several unique challenges, especially in the context of heterogeneity. Heterogeneity in data distributions, computational capabilities, and scenarios among clients necessitates the development of customized models and objectives in FL. Unfortunately, existing works such as FedAvg may not effectively accommodate the specific needs of each client. To address the challenges arising from heterogeneity in FL, we provide an overview of the heterogeneities in data, model, and objective (DMO). Furthermore, we propose a novel framework called federated mutual learning (FML), which enables each client to train a personalized model that accounts for the data heterogeneity (DH). A "meme model" serves as an intermediary between the personalized and global models to address model heterogeneity (MH). We introduce a knowledge distillation technique called deep mutual learning (DML) to transfer knowledge between these two models on local data. To overcome objective heterogeneity (OH), we design a shared global model that includes only certain parts, and the personalized model is task-specific and enhanced through mutual learning with the meme model. We evaluate the performance of FML in addressing DMO heterogeneities through experiments and compare it with other commonly used FL methods in similar scenarios. The results demonstrate that FML outperforms other methods and effectively addresses the DMO challenges encountered in the FL setting.

Key words: Federated learning; Knowledge distillation; Privacy preserving; Heterogeneous environment
https://doi.org/10.1631/FITEE.2300098 **CLC number:** TP39

1 Introduction

In the era of big data (Pan, 2017, 2018), the protection of data privacy is becoming increasingly important. This is not just a matter of public concern, but also a legal requirement enforced by laws such as the General Data Protection Regulation (GDPR) in the European Union. As a result, the massive

‡ Corresponding authors

* Project supported by the National Natural Science Foundation of China (Nos. U20A20387, 62006207, and 62037001), the Young Elite Scientists Sponsorship Program by China Association for Science and Technology (No. 2021QNRC001), the Zhejiang Provincial Natural Science Foundation, China (No. LQ21F020020), the Project by Shanghai AI Laboratory, China (No. P22KS00111), the Program of Zhejiang Province Science and Technology (No. 2022C01044), the StarryNight Science Fund of Zhejiang University Shanghai Institute for Advanced Study, China (No. SN-ZJU-SIAS-0010), and the Fundamental Research Funds for the Central Universities, China (Nos. 226-2022-00142 and 226-2022-00051)

ORCID: Chao WU, https://orcid.org/0000-0003-0885-6869; Fei WU, https://orcid.org/0000-0003-2139-8807

amounts of data generated by devices (e.g., mobile phones, wearables, and Internet of Things) (Liu PX et al., 2022) or organizations (e.g., hospitals, companies, and courts) (Padhya and Jinwala, 2019) cannot be collected in a central server, presenting a major challenge for deep learning (Li JH, 2018; Wu JX et al., 2018; Wang J et al., 2020). To address this challenge, federated learning (FL) has emerged as a novel deep learning setting (McMahan et al., 2017). FL enables clients to collaboratively train a shared model under the orchestration of a central server, while keeping the data decentralized (Corchado et al., 2016; Yang et al., 2019; Lim et al., 2020; Kairouz et al., 2021). This technique helps overcome the "data island" problem and has extensive applications, including mobile apps, autopilots, healthcare, and financial services. However, researchers face distinctive challenges when working on FL, particularly concerning heterogeneity. In this study, we focus on the heterogeneity problem and summarize it from three perspectives: data, model, and objective (DMO), as shown in Fig. 1. These challenges are distinct from those encountered in traditional distributed machine learning.

1. Data heterogeneity (DH). In FL, the data collected from multiple clients are non-independent and identically distributed (non-IID) as opposed to centralized deep learning, where data are independent and identically distributed (IID). This implies that the patterns of data generated by different clients $(\mathcal{X}_1, \mathcal{Y}_1), (\mathcal{X}_2, \mathcal{Y}_2), \cdots, (\mathcal{X}_k, \mathcal{Y}_k)$ have diverse distribution with $(x, y) \sim \mathcal{P}_i(x, y) \neq \mathcal{P}_j$ for any $i, j = 1, 2, \cdots, k$. The statistical heterogeneity of data can contribute to significant accuracy reduction, especially during the model weight averaging phase. As pointed out in Zhao et al. (2022), this occurs when the averaged model weights diverge due to the underlying differences in data distributions.

2. Model heterogeneity (MH). In FL, the global model obtained through FedAvg by aggregating the weights of local models cannot be customized for various scenarios and tasks. Clients vary in their hardware capabilities, the way they represent their local data, and the tasks they perform. Due to these differences, each client requires a personalized model that is specifically designed for their unique needs. A variety of studies, such as Wu BC et al. (2019) and He et al. (2021), have highlighted this concern. Also, due to privacy issues, local models need to be protected from theft, because they are considered the private property of clients. Local models may contain sensitive information, and thus their privacy preservation is an important aspect of FL. Gao DS et al. (2020) and Liang et al. (2020) further discussed the challenges of developing local models with varying data representations. Smith et al. (2017) suggested that different clients may have distinct and diverse goals, which necessitates designing individualized models adapted to those specific objectives.

3. Objective heterogeneity (OH). In FL, OH has two aspects, referring to the existence of different objectives between the global model and local models in FL, as well as across different clients. In one aspect, the server aims to train a generalized model that can fit the joint distribution $\mathcal{P}_{\text{joint}}(x, y)$ for all clients, while each client intends to train personalized models that can fit their own distribution $\mathcal{P}_k(x, y)$. However, reconciling these distinct goals may sacrifice the personalization of clients, particularly when non-IID data are involved. On the other aspect, some clients may share similar features, but have varying tasks such as 10- or 100-class classification, limiting the effectiveness of FL approaches like FedAvg. It is essential to address these objective heterogeneities to improve the performance of FL.

In this study, we propose a novel paradigm, entitled federated mutual learning (FML), which aims to address the challenges posed by the three sources of heterogeneity encountered in the FL context, namely, data, model, and objective. To tackle the issue of DH, we allow each client being able to train a personalized model, tailored to their specific data. This deviates from the conventional methodology of training a generalized global model, and allows for a more individualized service for each client. To address the issue of MH, we deploy a "meme model" on each client, which is a copy of the global model. The purpose of this model is to serve as an intermediary between the generalized global model and the personalized model. To enable the transfer of knowledge between these two models on local data, we introduce a knowledge distillation technique known as deep mutual learning (DML) (Zhang Y et al., 2018), which can be implemented during local updates. To overcome the issue of OH, the shared global (meme) model is no longer a complete model but includes only certain parts (such as convolution layers). The personalized model, on the other hand,

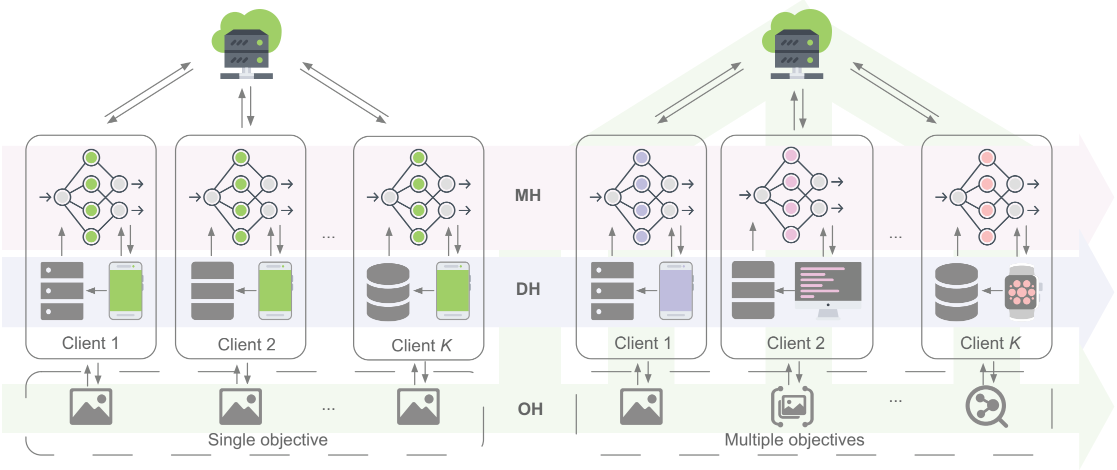

Fig. 1 The heterogeneities in FL can be divided into three types: DH, MH, and OH. DH refers to the fact that data generated by different clients are non-independent and identically distributed like in centralized deep learning. This statistical heterogeneity of data can result in significant accuracy reduction when model weights are averaged due to weight divergence. MH refers to the fact that clients may have different hardware capabilities, different representations of local data, or different tasks, and they need to design their own models. However, FedAvg cannot provide customized models for various scenarios and tasks, because it needs to aggregate the weights of local models with the same architecture. OH arises from the inconsistent objectives of the server and clients in FL. The server aims to construct a single generalized model from data contributed by all clients, while clients aim to train a personalized model for themselves. As a result, this trade-off between these two objectives can lead to the loss of both generalization and personalization. Additionally, clients may have data of similar features but different tasks, thereby complicating the model aggregation process. DH: data heterogeneity; FL: federated learning; MH: model heterogeneity; OH: objective heterogeneity

is designed to be task-specific and can be enhanced through mutual learning with the meme model. We conduct experiments to evaluate the performance of FML in addressing the challenges posed by DMO heterogeneities, as compared with other FL methods commonly used in similar scenarios. Our results demonstrate that FML outperforms the other methods and is highly effective in addressing the DMO challenges encountered in the FL setting.

2 Related works

2.1 Data heterogeneity

The key difference between FL and distributed learning, which typically refers to distributed training within data centers, lies in whether client data are locally fixed and inaccessible to others. This feature provides a safeguard for data privacy, but it results in non-IID and unbalanced data distribution, which complicates the training process. Non-IID data are difficult to train and can lead to reduced accuracy due to weight divergence, causing a considerable deviation from the correct weight updates during the averaging stage. Zhao et al. (2022) offered a data-sharing strategy to address this issue by creating a small, globally shared subset of data. This approach has proven effective in improving accuracy, and for privacy preservation, shared data can be extracted using distillation (Wang TZ et al., 2020) or be generated using a generative adversarial network (GAN) (Chen HT et al., 2019). Numerous theoretical works have focused on FedAvg, with a specific emphasis on convergence analysis and relaxing assumptions in the non-IID setting (Lian et al., 2017; Li X et al., 2019, 2021). However, it is essential to note that all these works concentrate on training a single global model. Zhang X et al. (2022) proposed a novel personalized FL method called pFedBayes that addresses the challenges of model overfitting and lack of statistical diversity among clients in FL. By introducing weight uncertainty and personalization through local distribution parameters, pFedBayes achieved better generalization error and convergence rates.

2.2 Model heterogeneity

In the context of FL, Smith et al. (2017) introduced the MOCHA framework, which addresses high communication costs, stragglers, and fault tolerance in multi-task FL. Similarly, Khodak et al. (2019) presented the average regret-upper-bound analysis (ARUBA) theoretical framework, which is used to analyze gradient-based meta-learning, and allows for training of separate models while maintaining control over model architectures by the central server. Li DL and Wang (2019) proposed a decentralized framework for FL based on knowledge distillation. This approach enables FL for independently designed models, but requires access to a public dataset and does not support a global model for future use. Additionally, this method does not support new participation since new participants may disrupt established models. Alam et al. (2023) proposed FedRolex, a model-heterogeneous FL approach that enables partial training and allows for the training of a global server model larger than the largest client model. FedRolex employs a rolling sub-model extraction scheme to mitigate client drift and outperforms state-of-the-art methods across models and datasets.

2.3 Objective heterogeneity

In traditional FL, the goal is to train a global model that is applicable to all clients. However, in personalized situations, Yu et al. (2022) demonstrated that some participants may not benefit from the global model when it is less accurate than their local model. The global model can become overfitted to the small local dataset, which impacts its personalization ability. As noted by Jiang et al. (2023), optimizing only for global accuracy can make the model more difficult to personalize. To achieve effective personalization in FL, Jiang et al. (2023) proposed three objectives: (1) developing improved personalized models that benefit most clients, (2) creating an accurate global model that benefits clients with limited private data for personalization, and (3) achieving rapid model convergence in a small number of training rounds. In the context of image representation in FL, Liu FL et al. (2020) proposed a framework for obtaining various image representations from different tasks and combining useful features from different vision-and-language grounding problems. Chen HY and Chao (2022) proposed FedRod, which addresses the dilemma of prioritizing a model's generic performance or personalized performance. By decoupling a model's dual duties with two prediction tasks, this framework can approach both goals simultaneously. This paper also demonstrated that the averaging over model weights acts as a regularizer for local models to improve their individual personalized performance.

3 Preliminaries

3.1 Typical federated learning setup

The main objective of typical FL, specifically the FedAvg algorithm, is to train a single shared model over decentralized data by minimizing the global objective function, $\min f(w)$, in a distributed manner. This function considers the entire dataset, which is the union of all decentralized data, and the loss function is over all private data, $f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$. Each client generates private data, denoted as $(\mathcal{X}_k, \mathcal{Y}_k)$, through a distinct distribution $\mathcal{P}_k(x, y)$ from K clients. To begin the FL process on each client, the weight vector $\boldsymbol{w}^k \in \mathbb{R}^d$ is copied from the global model. Each client then conducts a local update by optimizing the local objective using a gradient descent method for several epochs:

$$\begin{aligned} &F_k(\boldsymbol{w}^k) = \tfrac{1}{n_k}\textstyle\sum_{i\in\mathcal{P}_k} f_i(\boldsymbol{w}^k), \\ &\boldsymbol{w}^k \leftarrow \boldsymbol{w}^k - \eta\nabla F_k\left(\boldsymbol{w}^k\right), \end{aligned} \tag{1}$$

where $F_k(\boldsymbol{w}^k)$ represents the loss function of the k^{th} client, n_k represents the number of local samples, η is the learning rate, and $\nabla F_k\left(\boldsymbol{w}^k\right) \in \mathbb{R}^d$ is the gradient of $F_k\left(\boldsymbol{w}^k\right)$. It is important to note that the expectation $\mathbb{E}_{\mathcal{P}_k}\left[F_k(w)\right] = f(w)$ may not hold because $\mathcal{P}_k \neq \mathcal{P}_{\text{joint}}$ in the non-IID setting. Following a period of local updates, clients transmit local model weights, $\boldsymbol{w}^k$, to the parameter server, which then aggregates these weights by weighted averaging:

$$\boldsymbol{w}^{\text{global}} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n}\boldsymbol{w}^k, \tag{2}$$

where the aggregated weights, denoted as $\boldsymbol{w}^{\text{global}}$, represent the weights of the global model, and n denotes the number of samples over all clients. The entire training process is repeated until the global model achieves convergence. Through collaborative training, the shared global model can learn without the sharing of private local data. However, as previously mentioned, training local models directly on a copy of the global model presents challenges. To address these challenges, it is natural to train distinct models for clients.

3.2 Knowledge distillation

In the context of machine learning, knowledge distillation, as described by Hinton et al. (2015), is a process of transferring "dark knowledge" from a powerful, large teacher model to a lighter, easier-to-deploy student model, with minimal loss in performance. The loss function for a student model can be expressed in a simplified form as follows:

$$\left\{\begin{array}{l} L_{\text{student}} = L_{\text{CE}} + D_{\text{KL}}\left(p_{\text{teacher}} \| p_{\text{student}}\right), \\ p_{\text{teacher}} = \frac{\exp(z/T)}{\sum_i \exp(z_i/T)}, \end{array}\right. \tag{3}$$

where the loss function for a student model can be expressed as a combination of the cross entropy and Kullback Leibler (KL) divergence, denoted as L_{CE} and D_{KL}, respectively. The predictions of the teacher and student models are denoted as p_{teacher} and p_{student} respectively, parameter T represents the temperature hyperparameter, z_i represents the score of the i^{th} class, and z refers to the logits of the teacher model. By using the prediction of the teacher model, this method can improve the performance of the student model as it provides more useful information (soft targets) than the traditional one-hot label (hard targets), which can serve as a regularizer.

In this work, we incorporate knowledge distillation into the FL process during the local update stage. There are two main reasons for this approach: first, FL can be considered as a type of transfer learning between global and local models; second, the two models that transfer knowledge can have different architectures. However, as a well-trained teacher model is not readily available in the FL setting, we adopt DML, which is a deep learning strategy derived from knowledge distillation, for the local update process. Unlike the traditional teacher-to-student knowledge transfer pattern, DML is a two-way knowledge transfer, where both models can learn from each other throughout the training process. The loss function for the two models is expressed as follows:

$$\left\{\begin{array}{l} L_{w^1} = L_{\text{C1}} + D_{\text{KL}}\left(p_2 \| p_1\right), \\ L_{w^2} = L_{\text{C2}} + D_{\text{KL}}\left(p_1 \| p_2\right), \end{array}\right. \tag{4}$$

where the loss function for the two models in the DML strategy is expressed as a combination of the predictions p_1 and p_2 from the respective networks. The objective of DML is for the two models to train themselves over the dataset while achieving a consensus on predictions (i.e., distillation). This approach can result in better performance than independent training, and importantly, the two models can have different architectures, with the direction of knowledge transfer being two-way. Therefore, DML can be used to train distinct models during the local update stage of FL.

4 Methodology

4.1 Rethinking federated learning

FL faces three heterogeneities: data, model, and objective. To address these challenges, we propose rethinking two fundamental questions in FL: what is the product of FL and what should be shared in FL? In typical FL, the objective is to train a single model that fits a joint distribution $\mathcal{P}_{\text{joint}}(x, y)$ and can be used by all clients. However, in the context of OH, the server and clients have different objectives, with the server aiming to train a generalized model that fits $\mathcal{P}_{\text{joint}}(x, y)$ and clients seeking a personalized model that fits $\mathcal{P}_k(x, y)$. The non-IIDness of data (DH) presents challenges to training, but it can be beneficial for clients if it is possible to train a personalized model in FL. Hence, the non-IIDness of data should no longer be viewed as a bug but as a feature that enables clients to be served better personally. The model shared in FL does not necessarily have to be a complete, end-to-end (E2E) model. Instead, the model trained by FL can be split into two parts: a partial model that is shared globally and a partial model that is owned by clients locally, depending on what clients want to learn and share in FL. The shared objective can be an encoder for learning representations, a decoder for classification, or an integrated module for multi-task learning. Different local objectives (OH) of clients can lead to the need for MH, as clients may have similar but different tasks, such as visual question answering (VQA) and image captioning. Inspired by prior works (Gao DS et al., 2020; Li WH and Bilen, 2020; Liu FL et al., 2020; Gao JQ et al., 2023), we introduce DML as a way to address MH in FL. In DML, two models can learn from each other throughout the training process, which can result in better performance than independent training. Moreover, the two models can have different architectures, and the direction of knowledge transfer is two-way. Therefore, DML can be used to train distinct models during the local update stage of FL.

4.2 Federated mutual learning

To address the challenges presented by the three heterogeneities (DMO) in FL, it is necessary to train generalized and personalized models with different architectures. To this end, we introduce a more flexible FL method, named FML, in this subsection.

FL is a method that involves learning and transferring data knowledge between global and local models. To achieve this, we introduce a knowledge distillation approach known as DML, which is used as the local update method for clients to train a personalized model for their own data and task. Each client in FML has two models: the meme model that serves as the medium of knowledge transfer between global and local models, and the personalized model that is designed by clients for their specific data and task (Fig. 2). This allows clients to train their local model mutually with the global model rather than directly on it, thus making the process more flexible.

During the training process of FML, the global model is initialized and controlled by the central server, while each client initializes an initial personalized model customized for its own data and task. All clients then fork the global model as the meme model and conduct local updates, with the meme model constructed by splicing the forked global model with an adaptor layer if the global model is not a complete model. Rather than training directly on the copy of the global model, each client's local update involves DML between the meme model and personalized model for several epochs. The loss function of the two models can be rewritten as

$$L_{\text{local}} = \alpha L_{C_{\text{local}}} + (1-\alpha) D_{\text{KL}} \left(p_{\text{meme}} \| p_{\text{local}} \right), \quad (5)$$

$$L_{\text{meme}} = \beta L_{C_{\text{meme}}} + (1-\beta) D_{\text{KL}} \left(p_{\text{local}} \| p_{\text{meme}} \right), \quad (6)$$

where α and β are hyperparameters that control the relative importance of the cross-entropy loss L_{C} and the KL divergence term between the probability distributions of the local and meme models. α controls the weight given to the local model's cross-entropy loss $L_{C_{\text{local}}}$ versus the KL divergence term D_{KL} between the probability distributions of the meme and local models. A higher value of α puts more emphasis on the local model's accuracy in predicting labels, whereas a lower value of α places more emphasis on

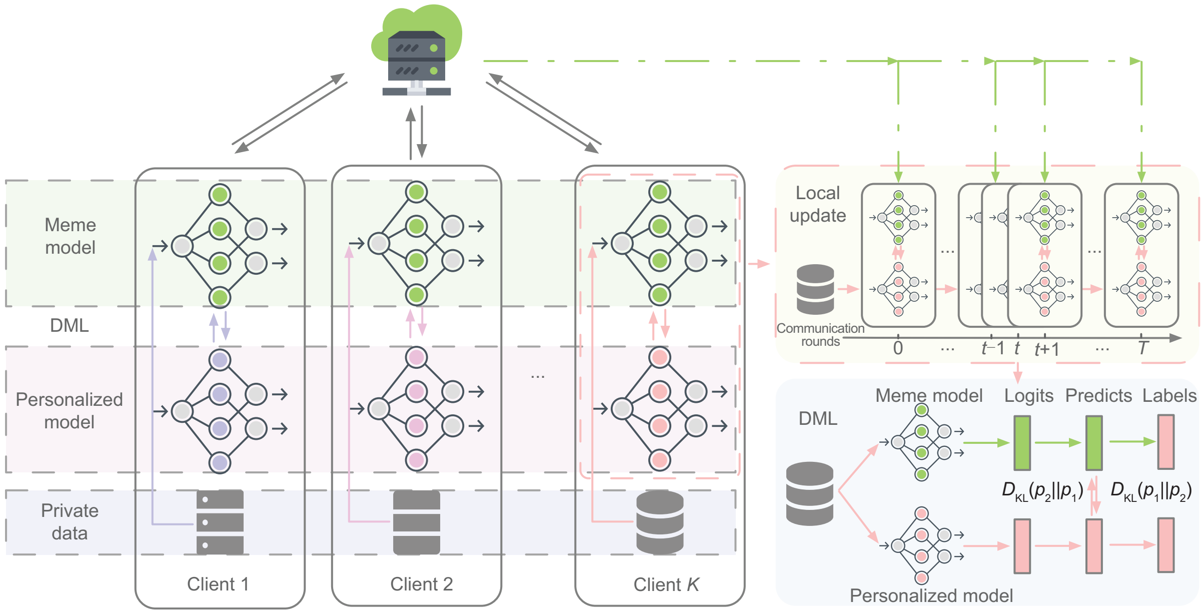

Fig. 2 In the FML method, each client trains two models over its private data during local update: the meme model and the personalized model. At each communication round, the clients fork the new generation of the global model as their meme model, while the personalized model is trained privately and continuously. During each local update, the two models in the clients engage in DML for several epochs, learning mutually. FML: federated mutual learning; DML: deep mutual learning

aligning the probability distributions of the meme and local models to improve the model's generalization ability. Similarly, β controls the weight given to the meme model's cross-entropy loss $L_{\mathrm{C}_{\mathrm{meme}}}$ versus the KL divergence term D_{KL} between the probability distributions of the local and meme models. A higher value of β puts more emphasis on the meme model's accuracy in predicting labels, whereas a lower value of β places more emphasis on aligning the probability distributions of the local and meme models to improve the model's generalization ability. The values of α and β need to be tuned carefully to balance the trade-off between the accuracy of the local and meme models and the alignment of their probability distributions. The mutual learning of the meme model and personalized model can be accomplished through DML. In this approach, the meme model transfers its knowledge to the personalized model by measuring the KL divergence of their outputs using the same input, denoted as Eq. (5). Likewise, the personalized model can convey feedback to the meme model to enhance its generalization, represented as Eq. (6). In addition, both models can be trained concurrently to facilitate mutual improvement. The direction of knowledge transfer is bidirectional, wherein the meme model shares its global knowledge with the personalized model and receives feedback from it, both of which are trained on private data. The trained meme models of individual clients are subsequently transmitted to the server, which averages them to obtain the new generation of global models. The entire process is reiterated until convergence is achieved, as outlined in Algorithm 1.

The global model (global), the meme model (meme), and the personalized model (local), along with private data $(\mathcal{X}, \mathcal{Y})$, are involved in executing the FML algorithm. In Algorithm 1, the subscripts and superscripts denote the t^{th} communication round and the k^{th} client, respectively. The maximum values of t, k, and e are represented by the upper case letters T, K, and E, respectively. It is important to note that our approach is distinct from typical FedAvg, because we abandon the weighted average item n_k/n and allow FML to degrade into FedAvg if $\beta = 1$. Specifically, from the perspective of the server, the global model is learned using FedAvg with the meme models of clients. This global model represents a generalized model that fits the joint distribution $\mathcal{P}_{\mathrm{joint}}(x, y)$ over all data. Conversely, from the perspective of clients, the personalized models are continuously trained on private data, while distilling knowledge from meme models at each communication round, as shown in Fig. 2. Importantly, throughout the entire process, the personalized models remain with the clients and are never replaced, thereby fitting the personalized distribution $\mathcal{P}_k(x, y)$ over private data. Further discussion of this approach is presented in Section 6.

Algorithm 1 Federated mutual learning

```
Server execution:
for each round $t = 1, 2, \cdots, T$ do
    for each client $k$ in parallel do
        $\mathrm{meme}_{t+1}^k \leftarrow \mathrm{ClientUpdate}(\mathrm{meme}_t^k)$
    end for
    Merge: $\mathrm{global}_{t+1} \leftarrow \frac{1}{K}\sum_{k=1}^{K} \mathrm{meme}_{t+1}^k$
end for
ClientUpdate:
for each client $k$ do
    Fork: $\mathrm{meme}_0^k \leftarrow \mathrm{global}_0$
    for each epoch $e = 1, 2, \cdots, E$ do
        Conduct DML between $\mathrm{meme}_t^k$ and $\mathrm{local}_t^k$ over private data $(\mathcal{X}_k, \mathcal{Y}_k)$
    end for
end for
```

In our FML framework, when participating clients require a service, they rely on their locally stored personalized model to make predictions. This personalized model is constructed based on the client's distinct distribution, which captures distinct features of each individual user. In contrast, when novel clients request a service, we deploy a global model. Although adopting a model from similar clients may appear attractive, it remains challenging to determine the similarity of two clients' data before accessing it. Thus, we opt for a more cautious approach by employing the global model to ensure optimal performance and preserve the privacy of all our clients' data.

5 Experiments

This section presents a comprehensive evaluation of the efficacy of FML over three frequently utilized image classification datasets, under both IID and non-IID conditions, using PyTorch. The experimental design comprises two main parts: the performance of FML is validated under typical FL settings, followed by an assessment of its performance under DMO settings.

5.1 Experimental settings

1. Datasets. In this study, three popular datasets, namely MNIST, and CIFAR-10, and CIFAR-100, are used to evaluate the effectiveness of FL. The MNIST dataset (LeCun et al., 1998) comprises 10 classes of handwritten digits ranging from 0 to 9, and it includes a total of 60 000 training images and 10 000 test images. Specifically, each digit has 6000 and 1000 images with a resolution of 28 × 28 pixels, allocated for training and test, respectively. The CIFAR-10/100 datasets (Krizhevsky, 2009) are 10- and 100-class classification datasets, respectively. The CIFAR-10 dataset contains 50 000 training images and 10 000 test images in 10 different classes, with 5000 and 1000 images per class, respectively. In contrast, the CIFAR-100 dataset has the same total number of images as CIFAR-10, but comprises 100 classes, with 500 training images and 100 test images per class. All images in CIFAR-10/100 are three-channel color images with a size of 32×32 pixels. These three datasets are commonly adopted in FL experiments.

2. Federated settings. In this study, we conduct experiments using a simulated cross-silo FL environment, in which a central server orchestrates the activities of five clients ($K = 5$). For the experiments of DH and MH, the total number of communication rounds on MNIST and CIFAR-100 datasets is set to $T = 200$, and T is set to 400 on the CIFAR-10 dataset. For the experiments of OH, the total number of communication rounds is set to $T = 250$. All the local epochs are set to $E = 5$. To ensure that the clients have equal amounts of training data in both IID and non-IID settings, the dataset is partitioned into five parts, with each client receiving 1/5 of the training data and 1/5 of the test data allocated as private validation data (for instance, CIFAR-10 comprises 50 000 training images and 10 000 test images, and hence, each client is assigned 10 000 images as private training data and 2000 images as private validation data in the IID setting without replacement). The overall test set, which includes 10 000 test images in CIFAR-10, is used to test the global model. In the IID setting, each client receives a set of shuffled data such that $\mathcal{P}_k(x,y) = \mathcal{P}_{\text{joint}}(x,y)$. However, in the non-IID setting, the dataset is divided into Kp shards of size $\frac{n}{Kp}$ (where $K = 5$ and $n = 50\ 000$), and p shards are assigned to each client. We consider three levels of non-IID difficulty, where p is set to $\{6,4,2\}$ for MNIST and CIFAR-10, and $\{60,40,20\}$ for CIFAR-100, which means that each client has a maximum of p classes of data. We denote these settings as non-IID (1, 2, 3), where non-IID (3) is an extreme setting with no overlap of classes.

3. Training settings. In our experimental setup, we evaluate the performance of four different models: multi-layer perceptron (MLP) (McMahan et al., 2017), LeNet5 (LeCun et al., 1989), a convolutional neural network (CNN1) that contains two 3×3 convolution layers (the first with six channels and the second with 16 channels), each followed by a 2×2 max pooling layer and rectified linear unit (ReLU) activation, and two fully connected (FC) layers, and a convolutional neural network (CNN2) that includes three 3×3 convolution layers, each with 128 channels, followed by a 2×2 max pooling layer and ReLU activation, and one FC layer. We use MLP and LeNet5 models for the MNIST dataset, and CNN1 and CNN2 models for CIFAR-10/100 datasets. The optimizer we select for all models is the stochastic gradient descent (SGD) algorithm, with a momentum of 0.9, weight decay of 5×10^{-4}, and batch size of 128.

5.2 FML in typical FL settings

Initially, we investigate the efficacy of FML in conventional FL settings, where all clients collaborate to train a common global model. To begin the experiments, we design and implement identical architectures for all models, including the global model, meme model, and personalized model for each client. We compare FML with two baselines, FedAvg (McMahan et al., 2017) and FedProx (Li T et al., 2020), in both IID and non-IID settings. FedProx is a typical FL method that aims to address the issue of heterogeneity in the devices' data distributions. It includes a regularization term called "proximal term" that is added to the loss function to encourage the model to be closer to a weighted average of the local models trained on each device. This helps mitigate the impact of the devices' different data distributions on the shared model's performance. We evaluate the accuracy of four different types of models (MLP, LeNet5, CNN1, and CNN2) over three datasets (MNIST, CIFAR-10, and CIFAR-100) in four data settings (IID, non-IID (1, 2, 3)), and report the accuracy of the global model in Table 1. We observe a decrease in the global model accuracy with the increasing level of difficulty in data setting from the top to the bottom of the table when comparing IID with non-IID (1, 2, 3). Additionally, comparing FML with baselines, we find that FML outperforms FedAvg and FedProx in most settings. We demonstrate the training process in Fig. 3.

5.3 FML in DMO

1. DH. Because the data in non-IID settings may be distributed differently across clients, the global shared model may not perform as well as local models trained solely on private data. Therefore, we address the challenge of DH by training a personalized model for each client. Personalized models allow each client

Table 1 Top-1 accuracies of global models in typical FL settings

Setting	Method	Accuracy (%)					
		MNIST		CIFAR-10		CIFAR-100	
		MLP	LeNet5	CNN1	CNN2	CNN1	CNN2
IID	FedAvg	98.44	99.29	85.90	**87.49**	56.11	60.88
	FedProx	98.14	99.13	83.91	86.15	32.41	59.23
	FML (ours)	**98.49**	**99.37**	**85.93**	87.41	**57.11**	**62.50**
Non-IID (1)	FedAvg	97.40	98.92	80.41	82.64	53.77	57.76
	FedProx	97.35	98.75	77.46	80.88	47.83	55.60
	FML (ours)	**97.70**	**99.07**	**80.86**	**82.69**	**54.21**	**59.77**
Non-IID (2)	FedAvg	96.84	98.67	**78.85**	**81.17**	50.86	**56.82**
	FedProx	96.98	98.50	76.53	78.87	45.46	55.34
	FML (ours)	**97.00**	**98.71**	78.64	80.85	**52.92**	55.93
Non-IID (3)	FedAvg	90.46	96.45	**63.22**	64.12	41.48	50.36
	FedProx	80.03	87.55	58.07	62.01	41.29	49.51
	FML (ours)	**93.77**	**96.70**	62.42	**66.75**	**46.30**	**51.86**

Best results are in bold. FL: federated learning; FML: federated mutual learning

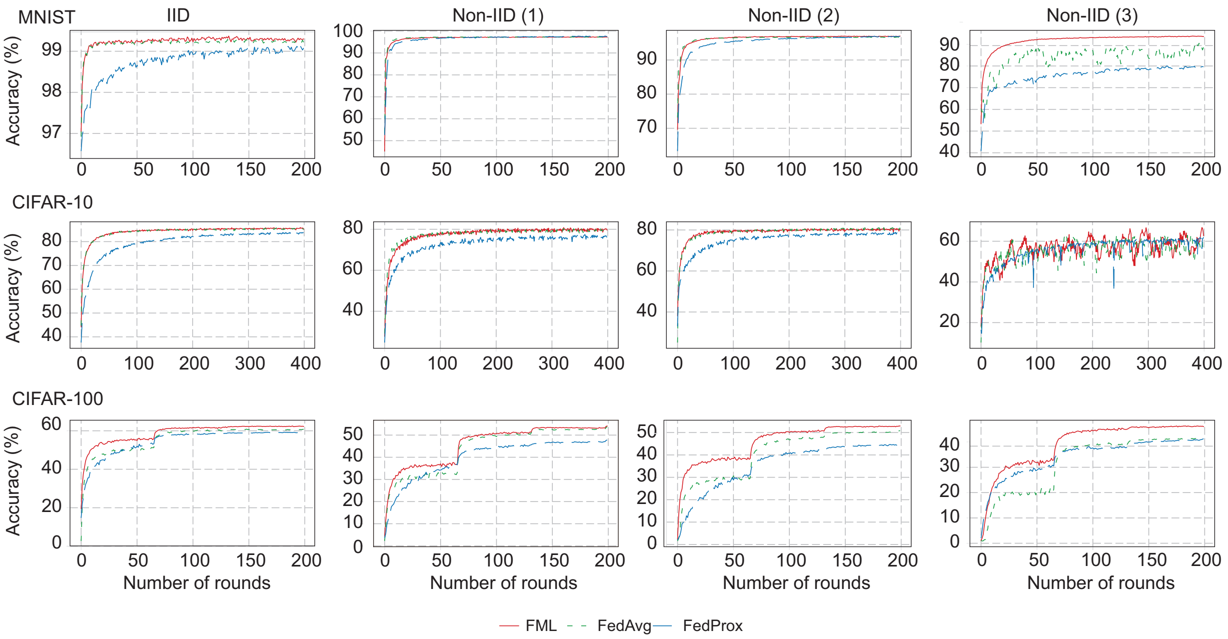

Fig. 3 Our findings indicate that FML offers better performance improvements than the FedAvg and FedProx approaches across four different data settings. We simulate various levels of DH, where the difficulty level gradually increases from left to right. With the aid of DML, the D_{KL} loss component acts as a powerful regularizer during the training process. In the non-IID setting, we observe that FML performs better with a stable trajectory compared to FedProx and FedAvg, which exhibit severe oscillations. As noted in Zhang Y et al. (2018), FML can identify a more stable and robust minimum. FML: federated mutual learning; DH: data heterogeneity; DML: deep mutual learning; non-IID: non-independent and identically distributed

to fit their own distribution $\mathcal{P}_k(x, y)$, rather than relying on a single generalized model, as is typically used in FL settings. We evaluate the efficacy of the product of FL, i.e., the personalized model in FML and the global model in FedAvg and FedProx, by measuring the performance on the private validation set. The results of this evaluation are presented in Fig. 4.

2. MH. To address the challenge of MH, we employ a knowledge distillation technique, which allows each client to design their personalized model based on specific requirements. In our experimental setup, we assign different models to each of the five clients, including one client using MLP, one employing LeNet5, one leveraging CNN1, and two making use of CNN2, and the global model is LeNet5. We train these models over CIFAR-10 in an IID setting and report the results in Fig. 5.

3. OH. We investigate multi-task FML, where clients may engage in different tasks. To this end, we initialize two clients to train 10-/100-class classification tasks over CIFAR-10/100 using LeNet5 and CNN1, respectively. The global model comprises only the convolution layers of CNN2, and it is not a complete E2E model. Therefore, an adaptor layer (an FC layer) must be appended to adapt to the 10-/100-class classification tasks of each client. The results of this study are presented in Fig. 6.

6 Discussion

1. Catfish effect. The ability of FML to handle MH allows clients to train models with varying dimensions and architectures, which may result in differences in model capabilities across clients. In our experiments, we observed a novel phenomenon, the catfish effect, which does not occur in FedAvg. This effect describes a scenario in which models with low capabilities (sardines) can be improved by a high-capability model (catfish), compared to the performance of only sardines in FML. Conversely, if a poorly trained model exists in FML, it has little effect

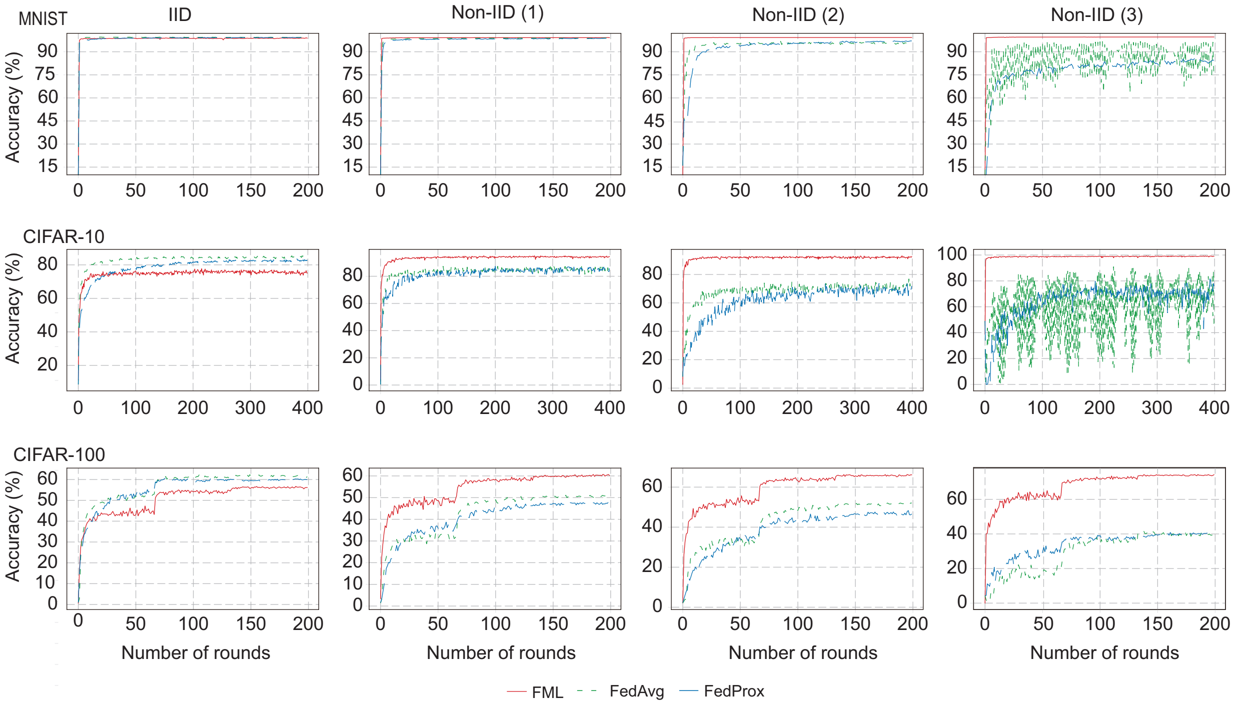

Fig. 4 The global model trained in an FL system functions as a generalized model, which exhibits poor performance on private data in non-IID settings. Through our analysis of the three curves, we observe that FedAvg exhibits more severe oscillations as the level of DH increases, particularly in non-IID (3). Although FedProx adds a proximal term to alleviate the oscillation, it fails to achieve a high accuracy. In contrast, FML rapidly improves and stabilizes at a high level, demonstrating superior performance in terms of both stability and accuracy. FL: federated learning; non-IID: non-independent and identically distributed; DH: data heterogeneity; FML: federated mutual learning

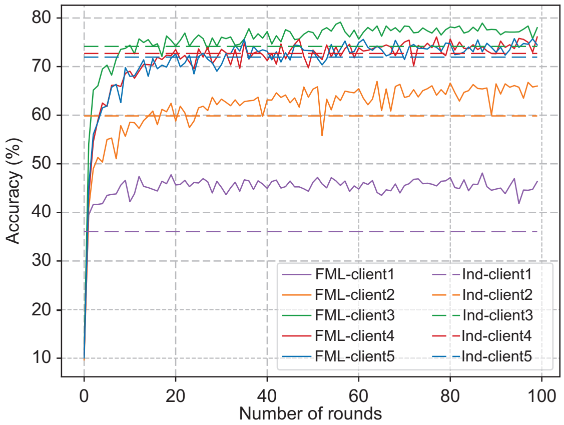

Fig. 5 We evaluate the performance of personalized models trained using FML (represented by the solid curves) and compare it to the highest accuracy achieved by the personalized models through independent training (represented by the dashed lines), using the private validation set. We show the first 100 rounds of the training process. Our results indicate that the use of a shared model through FML leads to improved accuracy for personalized models across all clients, regardless of the specific model architecture employed. FML: federated mutual learning

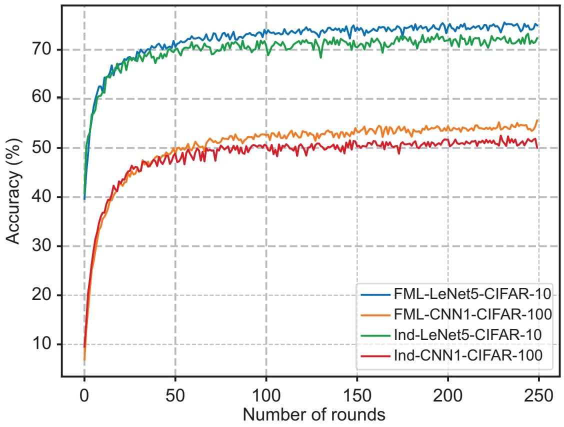

Fig. 6 We illustrate the performance of LeNet5 and CNN1 models, which were trained independently on CIFAR-10 and CIFAR-100 datasets, using the green and red curves, respectively. The blue and orange curves represent the performances of the two models trained using FML. We show the first 250 rounds of the training process. Our results demonstrate that the use of a shared representation through FML can effectively improve the accuracy of all models, despite the presence of different tasks assigned to each client. References to color refer to the online version of this figure

on the overall performance of other clients. This feature may inspire research on adversarial training in FL in the future.

2. Dynamic α and β. In our experiments, we fixed the proportions of cross-entropy loss and KL loss for both the local (α) and meme (β) models. However, we observed that dynamic α and β at different stages of training can significantly improve both the global and local model performances. Based on our experience, the improvement of the local model can be attributed to a well-trained global model at a later stage of training, while the improvement of the global model can be attributed to well-trained local models at an early stage of training. Therefore, a larger α in the early stage and a larger β in the later stage are preferred.

3. Privacy and fairness. In this study, we introduce the concept of model privacy. Because FML allows customized models which are the private property of individuals, it is crucial to protect the local customized models from theft. Furthermore, we have abandoned the use of the average item n_k/n in Section 4.2 due to privacy and fairness considerations. On one hand, the number of samples n_k on each client should not be exposed to the central server, because it could be used by attackers to breach privacy. On the other hand, different n_k values may lead to fairness issues, since clients with a larger number of samples would have a disproportionate influence on model training, which is not appropriate in some applications. Therefore, we have chosen to abandon this item and treat each client as equal, rather than each sample.

7 Conclusions

In this paper, we propose a novel federated mutual learning (FML) framework that effectively addresses the challenges of data, model, and objective (DMO) heterogeneities in federated learning (FL). By leveraging personalized models, meme models, and a deep mutual learning technique, FML provides a flexible approach for knowledge distillation between global and local models. The experimental results indicate that FML outperforms alternatives in different FL scenarios, thus establishing its effectiveness in dealing with DMO challenges. This study paves the way for more efficient and personalized learning strategies in the federated learning landscape.

Contributors

All authors contributed to the study conception and design. Tao SHEN, Fengda ZHANG, and Chao WU proposed

the motivation of the study. Tao SHEN, Jie ZHANG, and Xinkang JIA designed the method. Tao SHEN, Jie ZHANG, and Zheqi LV performed the experiments. Tao SHEN drafted the paper. All authors commented on previous versions of the paper. Kun KUANG, Chao WU, and Fei WU revised the paper. All authors read and approved the final paper.

Compliance with ethics guidelines

Fei WU is an editorial board member of *Frontiers of Information Technology & Electronic Engineering*. Tao SHEN, Jie ZHANG, Xinkang JIA, Fengda ZHANG, Zheqi LV, Kun KUANG, Chao WU, and Fei WU declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are openly available in public repositories. The MNIST dataset used in this study is publicly available and can be downloaded from the MNIST website (`http://yann.lecun.com/exdb/mnist/`). The CIFAR-10/100 datasets used in this study are also publicly available and can be downloaded from the CIFAR website (`https://www.cs.toronto.edu/~kriz/cifar.html`).

References

Alam S, Liu LY, Yan M, et al., 2023. FedRolex: model-heterogeneous federated learning with rolling sub-model extraction. https://arxiv.org/abs/2212.01548

Chen HT, Wang YH, Xu C, et al., 2019. Data-free learning of student networks. IEEE/CVF Int Conf on Computer Vision, p.3513-3521. https://doi.org/10.1109/ICCV.2019.00361

Chen HY, Chao WL, 2022. On bridging generic and personalized federated learning for image classification. https://arxiv.org/abs/2107.00778

Corchado JM, Li WG, Bajo J, et al., 2016. Special issue on distributed computing and artificial intelligence. *Front Inform Technol Electron Eng*, 17(4):281-282. https://doi.org/10.1631/FITEE.DCAI2015

Gao DS, Ju C, Wei XG, et al., 2020. HHHFL: hierarchical heterogeneous horizontal federated learning for electroencephalography. https://arxiv.org/abs/1909.05784

Gao JQ, Li JQ, Shan HM, et al., 2023. Forget less, count better: a domain-incremental self-distillation learning benchmark for lifelong crowd counting. *Front Inform Technol Electron Eng*, 24(2):187-202. https://doi.org/10.1631/FITEE.2200380

He CY, Annavaram M, Avestimehr S, et al., 2021. FedNAS: federated deep learning via neural architecture search. https://arxiv.org/abs/2004.08546v1

Hinton G, Vinyals O, Dean J, 2015. Distilling the knowledge in a neural network. https://arxiv.org/abs/1503.02531

Jiang YH, Konečný J, Rush K, et al., 2023. Improving federated learning personalization via model agnostic meta learning. https://arxiv.org/abs/1909.12488

Kairouz P, McMahan HB, Avent B, et al., 2021. Advances and open problems in federated learning. *Found Trends® Mach Learn*, 14(1-2):1-210. https://doi.org/10.1561/2200000083

Khodak M, Balcan MF, Talwalkar A, 2019. Adaptive gradient-based meta-learning methods. https://arxiv.org/abs/1906.02717

Krizhevsky A, 2009. Learning Multiple Layers of Features from Tiny Images. Master Thesis, Department of Computer Science, University of Toronto, Canada.

LeCun Y, Boser B, Denker J, et al., 1989. Handwritten digit recognition with a back-propagation network. Proc 2nd Int Conf on Neural Information Processing Systems, p.396-404.

LeCun Y, Bottou L, Bengio Y, et al., 1998. Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11):2278-2324. https://doi.org/10.1109/5.726791

Li DL, Wang JP, 2019. FedMD: heterogenous federated learning via model distillation. https://arxiv.org/abs/1910.03581

Li JH, 2018. Cyber security meets artificial intelligence: a survey. *Front Inform Technol Electron Eng*, 19(12):1462-1474. https://doi.org/10.1631/FITEE.1800573

Li T, Sahu AK, Zaheer M, et al., 2020. Federated optimization in heterogeneous networks. https://arxiv.org/abs/1812.06127v5

Li WH, Bilen H, 2020. Knowledge distillation for multi-task learning. Proc European Conf on Computer Vision, p.163-176. https://doi.org/10.1007/978-3-030-65414-6_13

Li X, Huang KX, Yang WH, et al., 2019. On the convergence of FedAvg on non-IID data. https://arxiv.org/abs/1907.02189

Li X, Yang WH, Wang SS, et al., 2021. Communication efficient decentralized training with multiple local updates. https://arxiv.org/abs/1910.09126v1

Lian XR, Zhang C, Zhang H, et al., 2017. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. Proc 31st Int Conf on Neural Information Processing Systems, p.5336-5346.

Liang PP, Liu T, Liu ZY, et al., 2020. Think locally, act globally: federated learning with local and global representations. https://arxiv.org/abs/2001.01523

Lim WYB, Luong NC, Hoang DT, et al., 2020 Federated learning in mobile edge networks: a comprehensive survey. *IEEE Commun Surv Tutor*, 22(3):2031-2063. https://doi.org/10.1109/COMST.2020.2986024

Liu FL, Wu X, Ge S, et al., 2020. Federated learning for vision-and-language grounding problems. *Proc AAAI Conf Artif Intell*, 34(7):11572-11579. https://doi.org/10.1609/aaai.v34i07.6824

Liu PX, Jiang JM, Zhu GX, et al., 2022. Training time minimization for federated edge learning with optimized gradient quantization and bandwidth allocation. *Front Inform Technol Electron Eng*, 23(8):1247-1263. https://doi.org/10.1631/FITEE.2100538

McMahan B, Moore E, Ramage D, et al., 2017. Communication-efficient learning of deep networks from decentralized data. Proc 20th Int Conf on Artificial Intelligence and Statistics, p.1273-1282.

Padhya M, Jinwala DC, 2019. MULKASE: a novel approach for key-aggregate searchable encryption for multi-owner data. *Front Inform Technol Electron Eng*, 20(12):1717-1748. https://doi.org/10.1631/FITEE.1800192

Pan YH, 2017. Special issue on artificial intelligence 2.0. *Front Inform Technol Electron Eng*, 18(1):1-2. https://doi.org/10.1631/FITEE.1710000

Pan YH, 2018. 2018 special issue on artificial intelligence 2.0: theories and applications. *Front Inform Technol Electron Eng*, 19(1):1-2. https://doi.org/10.1631/FITEE.1810000

Smith V, Chiang CK, Sanjabi M, et al., 2017. Federated multi-task learning. Proc 31st Int Conf on Neural Information Processing Systems, p.4427-4437.

Wang J, Li R, Wang J, et al., 2020. Artificial intelligence and wireless communications. *Front Inform Technol Electron Eng*, 21(10):1413-1425. https://doi.org/10.1631/FITEE.1900527

Wang TZ, Zhu JY, Torralba A, et al., 2020. Dataset distillation. https://arxiv.org/abs/1811.10959

Wu BC, Dai XL, Zhang PZ, et al., 2019. FBNet: hardware-aware efficient ConvNet design via differentiable neural architecture search. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.10726-10734. https://doi.org/10.1109/CVPR.2019.01099

Wu JX, Li JH, Ji XS, 2018. Security for cyberspace: challenges and opportunities. *Front Inform Technol Electron Eng*, 19(12):1459-1461. https://doi.org/10.1631/FITEE.1840000

Yang Q, Liu Y, Cheng Y, et al., 2019. Federated Learning. Springer, Cham, Switzerland, p.1-207.

Yu T, Bagdasaryan E, Shmatikov V, 2022. Salvaging federated learning by local adaptation. https://arxiv.org/abs/2002.04758

Zhang X, Li YC, Li WP, et al., 2022. Personalized federated learning via variational Bayesian inference. Proc Int Conf on Machine Learning, p.26293-26310.

Zhang Y, Xiang T, Hospedales TM, et al., 2018. Deep mutual learning. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4320-4328. https://doi.org/10.1109/CVPR.2018.00454

Zhao Y, Li M, Lai LZ, et al., 2022. Federated learning with non-IID data. https://arxiv.org/abs/1806.00582