

# dFLMoE: Decentralized Federated Learning via Mixture of Experts for Medical Data Analysis

Luyuan Xie<sup>1,2,3\*</sup>  
Nan Xi<sup>4</sup>

Tianyu Luan<sup>1,3,4†</sup>  
Yuejian Fang<sup>1,2,3</sup>

Wenyuan Cai<sup>1</sup>  
Qingni Shen<sup>1,2,3</sup>

Guochen Yan<sup>2,3</sup>  
Zhonghai Wu<sup>1,2,3</sup>

Zhaoyu Chen<sup>1,2,3</sup>  
Junsong Yuan<sup>4</sup>

<sup>1</sup>School of Software and Microelectronics, Peking University

<sup>2</sup>PKU-OCTA Laboratory for Blockchain and Privacy Computing

<sup>3</sup>National Engineering Research Center for Software Engineering, Peking University

<sup>4</sup>State University of New York at Buffalo

## Abstract

Federated learning has wide applications in the medical field. It enables knowledge sharing among different health-care institutes while protecting patients' privacy. However, existing federated learning systems are typically centralized, requiring clients to upload client-specific knowledge to a central server for aggregation. This centralized approach would integrate the knowledge from each client into a centralized server, and the knowledge would be already undermined during the centralized integration before it reaches back to each client. Besides, the centralized approach also creates a dependency on the central server, which may affect training stability if the server malfunctions or connections are unstable. To address these issues, we propose a decentralized federated learning framework named dFLMoE. In our framework, clients directly exchange lightweight head models with each other. After exchanging, each client treats both local and received head models as individual experts, and utilizes a client-specific Mixture of Experts (MoE) approach to make collective decisions. This design not only reduces the knowledge damage with client-specific aggregations but also removes the dependency on the central server to enhance the robustness of the framework. We validate our framework on multiple medical tasks, demonstrating that our method evidently outperforms state-of-the-art approaches under both model homogeneity and heterogeneity settings.

## 1. Introduction

Federated learning has extensive medical applications. A well-designed federated system can protect data privacy while sharing high-level knowledge among different clients. This enables each client's network to receive additional sup-

\*This work was supported by the National Key R&D Program of China under Grant No.2022YFB2703301.

†Tianyu Luan is corresponding author: tianyulu@buffalo.edu

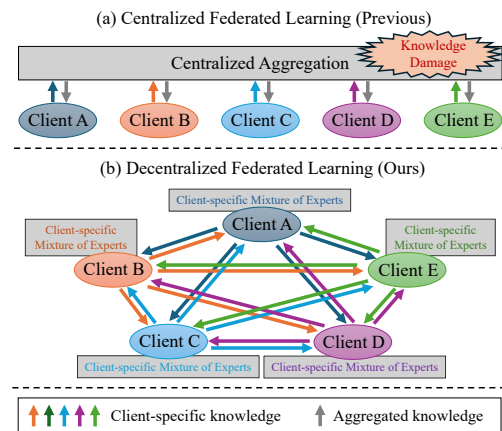


Figure 1. (a) Previous centralized federated learning framework aggregates knowledge from each client in a centralized server. This process can lead to knowledge damage in centralized aggregation and the framework is heavily dependent on the central server's stability. (b) Our decentralized framework dFLMoE eliminates centralized server and aggregation by having clients directly exchange knowledge with each other. Each client then uses a Mixture of Experts (MoE) approach to adaptively combine local and received knowledge.

port and achieve better performance and generalizability. In medical scenarios, patient data is hard to collect and has strong privacy protection requirements. Federated learning systems can effectively address the data limitations at each healthcare institution, enhancing their model performance and generalizability while ensuring privacy.

Existing federated learning systems, such as [8, 9, 27, 36], are designed in a centralized manner. In each training round, each client needs to upload client-specific knowledge (e.g. model parameters) to a central server for aggregation, which is then distributed back to each client. Regarding aggregation methods, they require a unified model structure [8, 9, 36], a centralized messenger model [52, 53, 56], or a unified public dataset [20, 23, 28, 58]. Such centralized designs achieve

good results, but this design may lead to performance bottlenecks. As shown in Figure 1(a), centralized federated learning frameworks, such as [36, 58], would distill knowledge from each client from their local data, and then send that knowledge to a centralized server which aggregates that knowledge into a single model. However, the aggregation process would typically merge the information from all client models into a single aggregated model, mostly with a simple merging scheme such as weighted sum. Considering the domain and data distribution differences among clients, the same aggregation process for all clients would result in potential knowledge damage even before the aggregated knowledge gets back to each client. Moreover, such centralized aggregation methods, particularly weighted sum schemes, are also widely used in federated systems like [21, 24, 36], which may not preserve the knowledge of each client well and could possibly hurt the performance of the federated learning framework. Furthermore, centralized federated frameworks heavily depend on the central server and the stability of its connections. If the central server malfunctions or the connections to it are unstable, the training stability of each client can be significantly affected.

To address the knowledge damage of centralized aggregation and to reduce the dependence on the centralized server, we propose a decentralized approach to design a federated learning framework. As illustrated in Figure 1(b), to minimize knowledge damage during model aggregation, we eliminate the centralized model aggregation operation. Instead, during the knowledge exchange process, the knowledge that each client would originally send to the server is now directly transmitted to other clients. This way, each client can receive the full knowledge sent by others without damage. Note that the communications between clients do not involve any patient data, which allows us to effectively protect patient privacy. Then, within each client, we design a Mixture of Experts (MoE) approach, treating the knowledge received from other clients and the client's own local knowledge as individual experts, and making decisions collectively using these experts. This decentralized design enables each client to consider its own local data and adaptively select the participation and weights of the experts and also avoids the unnecessary knowledge damage that occurs in centralized systems when aggregating into a unified model. Furthermore, it eliminates the reliance on a central server. If a client or some connections are unstable, the training of our framework would still be effective and without interruption.

Our decentralized system is named Decentralized Federated Learning via Mixture of Experts (dFLMoE). In each training round, we first train the local model of each client, which consists of a body and a head. Each client's body processes the input and encodes it into features, which are then passed through the head to obtain the final results. After local training, we send the model heads from each client to

all other clients. Considering that a decentralized framework requires model transmission between every pair of clients, transmitting only the lightweight head models would significantly reduce communication costs. After obtaining the heads from other clients, we train an attention-based MoE model, adaptively selecting the most effective combination of heads on each client to obtain the final results. Such client-specific MoE design does not require a structure consistency of the head from each client, which can effectively accommodate the commonly occurring model heterogeneity in practical medical scenarios. Moreover, due to the decentralized nature of the system, when a certain client encounters issues, other clients can still be trained without interruption. If the connection between two clients drops, the knowledge from these clients can still be shared through others, enhancing the robustness of the framework.

In summary, our contributions are as follows:

- We propose a decentralized federated learning framework named dFLMoE. Our framework directly transmits each client's knowledge to other clients and performs local decision-making on each client, effectively avoiding the knowledge damage caused by centralized server aggregation and eliminating the dependence on a central server.
- We design a lightweight Mixture of Experts (MoE) module for each client. This local MoE module can adaptively make client-specific decisions using lightweight experts from local and other clients, which can better adapt knowledge from other clients to improve performance and generalizability, without notably increasing communication costs.
- We validate the effectiveness of our framework on 5 different medical tasks. Our experimental results demonstrate that, on these tasks, under both model homogeneity and heterogeneity settings, our method evidently outperforms the state-of-the-art.

## 2. Related Works

**Centralized federated learning.** The general paradigm of federated learning involves clients uploading their local knowledge to a central server for aggregation, which is then distributed back to all clients. Based on the type of aggregation methods, this can be divided into three main categories: local model parameters aggregation [9, 21, 24, 25, 33, 36, 43], soft predictions aggregation [20, 23, 28, 58], and messenger model parameters aggregation [52, 53, 55, 56]. The framework for local model parameters aggregation requires aggregating all or part of the local model parameters at the central server [8, 14, 18, 25, 27, 35, 37, 48]. They require consistent local model structure [2, 7, 34, 51]. Federated learning frameworks that aggregate soft predictions require a public dataset, limiting their application in medical scenarios. Frameworks based on aggregating messenger model parameters insert a homogeneous model into each client and

share this model to transfer knowledge. These centralized approaches can lead to knowledge damage during aggregation, and the knowledge would be undermined before it reaches back to each client. Meanwhile, if the central server malfunctions or the connections are unstable, the training stability of each client can be significantly impacted.

**Decentralized federated learning.** Decentralized federated learning, also known as peer-to-peer federated learning [40], addresses the dependency on a central server. Currently, mainstream research in decentralized federated learning focuses on integrating it with blockchain to further enhance security and privacy [1, 30, 38, 59]. However, these works did not address the statistical heterogeneity and system heterogeneity issues in federated learning. Meanwhile, recent work [6, 26, 41, 47] has emerged to improve the performance of decentralized federated learning. They have only decentralized the security aspect without decentralizing the algorithm. Our method adopts a localized knowledge fusion approach, allowing us to adaptively select knowledge based on each client’s needs, thereby reducing knowledge damage.

### 3. Method

#### 3.1. Overview

We design a decentralized federated learning framework named dFLMoE to address the knowledge damage of centralized aggregation and reduce the dependence on the centralized server. Specifically, we firstly train a local network for client  $i$  by its private dataset  $D_i = \{x_i, y_i\}$ , where  $x_i$  is the input data in  $D_i$ , and  $y_i$  is the label. Then each client shares their learned knowledge  $K$  with other clients. Finally, we achieve the final decision through knowledge fusion using Mixture of Experts (MoE). The dFLMoE’s paradigm can be expressed as:

$$\mathbb{G} = \bigcup_{i=0}^N f_i(\theta_i; x_i; \{K_1, \dots, K_i, \dots, K_N\}), \quad (1)$$

where  $f_i(\theta_i; x_i; \{K_1, \dots, K_i, \dots, K_N\})$  is the model for client  $i$ , where  $\theta_i$  is the parameters of  $f_i$ ,  $x_i$  is model input,  $K_i$  is the knowledge shared by client  $i$ ,  $N$  represents the total number of participating clients, and  $\mathbb{G}$  represents the set of  $f_i$ .

The pipeline of dFLMoE is shown in Figure 2. In each client, the model includes the local network and invited experts. The local network is divided into four parts: Body, Feature space transform, Head, and Mixture of Experts (MoE). The body model is used to extract features. Feature space transform module converts the local features into the feature space corresponding to the respective experts (heads). The head model generates the network output using the features and the head module of each client is also shared among all clients, with the heads invited from other clients forming the

Mixture of Experts module for each client. We treat each head as an expert and use a Mixture of Experts (MoE) approach to get the final outputs. Our training process consists of 3 steps: a) Local network training, b) Sharing the local head among clients, and c) Mixture of Experts decision. In the rest of the section, we will explain each step in detail.

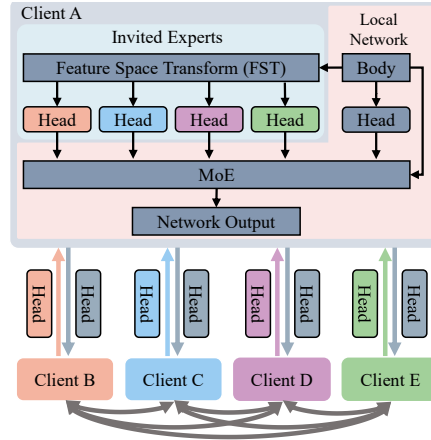


Figure 2. Overview of our proposed dFLMoE framework. For each training phase, we first train the Local network (Body and Head) while freezing the parameters of the MoE module (top right). Then, we send and receive the head to share knowledge among clients (bottom). Finally, we do a Mixture-of-Experts (MoE) decision by training the Feature space transform and MoE network while freezing other parameters including the local body and all the heads. More details can be found in the Sec. 3.

#### 3.2. Local Network Training

At this stage, our goal is to obtain the local network with local knowledge by local data. Therefore, we only train the head and body of the local network and freeze the parameters of the feature space transform and MoE module. For the client  $i$ , the local network output  $\hat{y}_i^l$  can be defined as:

$$\hat{y}_i^l = F_{h,i}(F_{b,i}(x_i)), \quad (2)$$

where  $F_{b,i}(\cdot)$  and  $F_{h,i}(\cdot)$  are the body and head of the local network in client  $i$ , respectively. The MoE output  $\hat{y}_i^m$  can be represented as:

$$\hat{y}_i^m = M\left(\bigcup_{j=1}^N F_{h,j}(FT_j(F_{b,i}(x_i))), F_{b,i}(x_i)\right), \quad (3)$$

where  $M(\cdot)$  is the MoE network and  $FT_j$  is the feature space transform of Experts  $j$  (See Sec. 3.4 for details) in Fig.2,  $F_{b,i}(\cdot)$  is the local model body,  $\bigcup_{j=1}^N F_{h,j}(FT_j(F_{b,i}(x_i)))$  is the set of predictions from each expert (head).  $N$  is the total number of participating clients, excluding the local client.

Finally, for client  $i$ , its training loss function  $\mathcal{L}_{ln,i}$  is:

$$\mathcal{L}_{ln,i} = \lambda_{loc} \mathcal{L}_{loc}(\hat{y}_i^l, y_i) + \lambda_{MoE} \mathcal{L}_{MoE}(\hat{y}_i^m, y_i). \quad (4)$$

$\mathcal{L}_{loc}$  and  $\mathcal{L}_{MoE}$  represent the loss functions of the local network and the MoE, respectively. For classification tasks,  $\mathcal{L}_{loc}$  and  $\mathcal{L}_{MoE}$  are cross-entropy loss. For super-resolution tasks,  $\mathcal{L}_{loc}$  and  $\mathcal{L}_{MoE}$  are  $L1$  loss. And for segmentation tasks,  $\mathcal{L}_{loc}$  and  $\mathcal{L}_{MoE}$  are Dice and cross-entropy loss.  $\lambda_{loc}$  and  $\lambda_{MoE}$  are their corresponding weights.  $y_i$  is the label of local data  $x_i$ . More details can be found in supplementary materials.

### 3.3. Localized knowledge Exchange

In the communication stage of existing decentralized federated learning, each of the  $N$  participating clients needs to share its local model with the other  $N - 1$  clients. Thus, in total there are  $N(N - 1)$  times of communication, which is significantly higher than the centralized federated learning, which would only need  $2N$  communications for both uploading and downloading. To reduce the communication cost of decentralized federated learning, in the Sharing local head among clients phase, we only share the head of the local model instead of the entire local model. The parameters of the head are several orders of magnitude smaller than those of the local model, which significantly reduces computational costs. Compared to centralized federated learning, this approach does not introduce a significant communication burden. Our experiments demonstrate that, in contrast to sharing the entire local model with each client, our communication overhead is only 0.02% of theirs, while our performance remains comparable to theirs.

### 3.4. Mixture of Experts Decision

This stage is designed to learn the combination weights of all experts based on the local data. During this stage of training, we fine-tune the parameters of feature space transform and MoE while freezing other parameters. The Mixture of Experts Decision loss function  $\mathcal{L}_{MD,i}$  for client  $i$  is defined as:

$$\mathcal{L}_{MD,i} = \mathcal{L}_{MoE}(\hat{y}_i^m, y_i), \quad (5)$$

where  $\mathcal{L}_{MoE}$  is the loss functions of the MoE. For classification tasks,  $\mathcal{L}_{MoE}$  is cross-entropy loss. For super-resolution tasks,  $\mathcal{L}_{MoE}$  is  $L1$  loss. And for segmentation tasks,  $\mathcal{L}_{MoE}$  is Dice and cross-entropy loss. During inference, we directly use the output of the MoE as the final prediction. The experimental results show that dFLMoE can be applied to federated learning scenarios with data heterogeneity, model homogeneity, and model heterogeneity without notably increasing communication costs.

**Feature space transform in MoE.** Before the final Mixture of Experts decision, we design a feature space transform module to transform the local features into the corresponding expert's feature space. As shown in Figure 3, the local

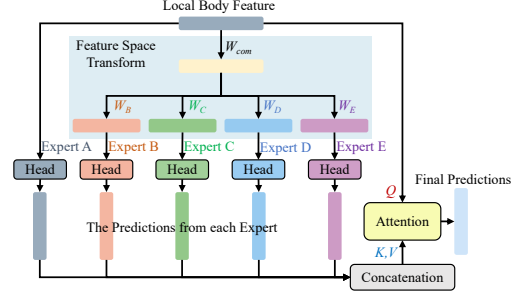


Figure 3. The structure of Mixture of Experts and Feature Space Transform. Firstly, the Feature Space Transform converts the local body feature into the feature space corresponding to each expert. Then, each feature obtains the final prediction through the respective expert, and we collect all predictions as the Key  $K$  and Value  $V$ . Next, we generate the query  $Q$  using the local body feature through a linear layer. Finally, we perform the attention mechanism with  $Q$ ,  $K$ , and  $V$  to obtain the final predictions.

body feature is first transformed into a common space by  $W_{com}$ , and then separately transformed into the corresponding expert's feature space through the respective  $W_j$ . In classification tasks,  $W_{com}$  and  $W_j$  are linear layers. For the segmentation and super-resolution tasks,  $W_{com}$  and  $W_j$  are convolutional layers.

After feature space transform, the features in the corresponding space generate the predictions through the respective experts. We utilize the Mixture of Experts (MoE) framework to effectively aggregate these predictions. To enhance the MoE's focus on key experts, inspired by [5, 29], we incorporate a cross-attention mechanism to learn the weights associated with the predictions generated by each client's experts. It is important to note that local models not only capture essential information from their respective local datasets but also tend to inherit biases that may contribute to overfitting. By utilizing the experts from all clients as candidates and employing local features as queries to extract relevant information from the collective pool of experts, we ensure that the selected information reflects the common knowledge shared across clients. We posit that this public information, derived from diverse datasets, possesses greater generalizability, while the local biases are effectively mitigated in the selection process. Consequently, we propose the adoption of a cross-attention design to filter out local biases and enhance the overall generalization capability of the model.

The MoE is illustrated in Figure 3. We involve the local body feature denoted as  $I$ , and concatenate all the expert predictions as  $K$  and  $V$ . The feature  $I$  obtains the Query feature  $Q$  through a linear layer  $W$ . The prediction of the MoE  $y_{MoE}$  is represented as:

$$y_{MoE} = Attention(W(I), K, V), \quad (6)$$

Table 1. The results of classification task in different resolutions with homogeneous models or heterogeneous models. The  $x2\downarrow$ ,  $x4\downarrow$ , and  $x8\downarrow$  are downsampling half, quarter, and eighth of high-resolution images. We evaluate ACC and MF1 results on the BreakHis dataset. The larger the better. **Bold** number means the best. The **red** boxes represent the single model federated learning and personalized federated learning methods, and their individual clients use the homogeneous model settings (ResNet5). The **blue** boxes represent the method of using heterogeneous models. The four client models are set to ResNet{17, 11, 8, 5}, respectively. In two different model settings, dFLMoE achieves the best performance.

Methods	HR		x2		x4		x8		Average	
	ACC $\uparrow$	MF1 $\uparrow$	ACC $\uparrow$	MF1 $\uparrow$	ACC $\uparrow$	MF1 $\uparrow$	ACC $\uparrow$	MF1 $\uparrow$	ACC $\uparrow$	MF1 $\uparrow$
Only Local Training	0.7491	0.6719	0.7568	0.6856	0.7015	0.6135	0.6956	0.5867	0.7258	0.6394
FedAvg	0.6067	0.4621	0.6667	0.5874	0.6178	0.5194	0.5799	0.4616	0.6178	0.5076
SCAFFOLD	0.6263	0.4821	0.7156	0.6597	0.6475	0.5906	0.5702	0.4969	0.6399	0.5573
FedProx	0.6195	0.4958	0.6862	0.6271	0.6467	0.5632	0.4664	0.3495	0.6047	0.5089
Ditto	0.7111	0.6557	0.7321	0.6404	0.7261	0.6743	0.6854	0.5932	0.7137	0.6409
APFL	0.6412	0.5848	0.6033	0.5626	0.7301	0.6468	0.6973	0.6166	0.6680	0.6027
FedRep	0.7663	0.7165	0.7513	0.6869	0.6849	0.6151	0.7254	0.6229	0.7320	0.6604
LG-FedAvg	0.7358	0.6504	0.7733	0.6726	0.7182	0.6323	0.7173	0.6481	0.7362	0.6509
MH-pFLID	0.8282	0.7762	0.8308	0.7829	0.8180	0.7674	0.7560	0.6933	0.8083	0.7550
dFLMoE (Ours)	<b>0.8652</b>	<b>0.8360</b>	<b>0.8597</b>	<b>0.8322</b>	<b>0.8423</b>	<b>0.8063</b>	<b>0.7602</b>	<b>0.7131</b>	<b>0.8319</b>	<b>0.7969</b>
Only Local Training	0.7891	0.7319	0.8027	0.7461	0.7538	0.6852	0.6956	0.5867	0.7603	0.6875
FedMD	0.7599	0.7083	0.8321	0.7829	0.7721	0.7293	0.6495	0.5439	0.7534	0.6911
FedDF	0.7661	0.7253	0.8132	0.7629	0.7826	0.7342	0.6627	0.5627	0.7562	0.6963
pFedDF	0.8233	0.7941	0.8369	0.7965	0.8121	0.7534	0.6843	0.6022	0.7892	0.7366
DS-pFL	0.7842	0.7609	0.8334	0.7967	0.7782	0.7258	0.6327	0.5229	0.7571	0.7016
KT-pFL	0.8424	0.8133	0.8441	0.8011	0.7801	0.7325	0.7032	0.6219	0.7925	0.7422
MH-pFLID	0.8929	0.8658	0.8992	0.8787	0.8661	0.8327	0.7751	0.7130	0.8583	0.8226
dFLMoE (Ours)	<b>0.9048</b>	<b>0.8898</b>	<b>0.9205</b>	<b>0.9064</b>	<b>0.9039</b>	<b>0.8865</b>	<b>0.8227</b>	<b>0.7819</b>	<b>0.8880</b>	<b>0.8662</b>

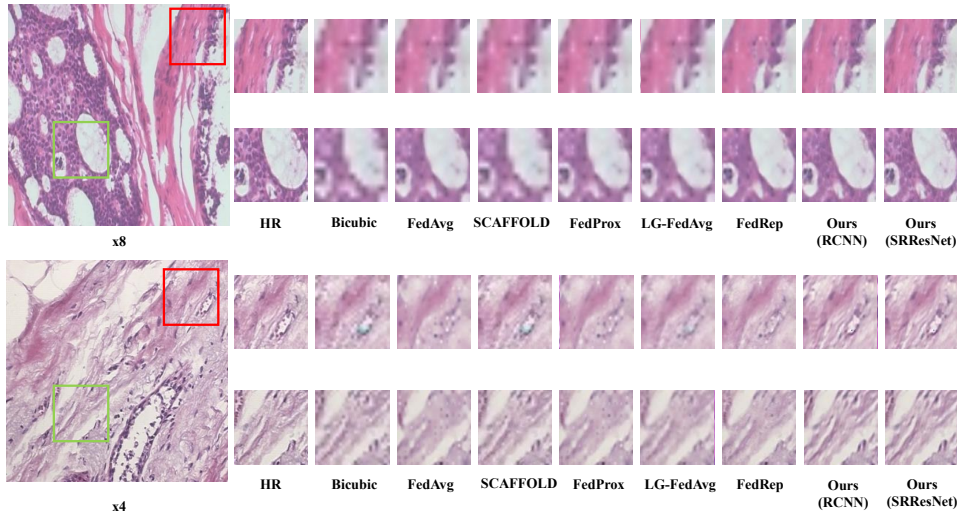


Figure 4. Visualized comparison of Federated Learning in medical image super-resolution. We randomly select two samples from different resolutions ( $x8\downarrow$  and  $x4\downarrow$ ) to form the visualization. Super-resolution results for FedAVG, SCAFFOLD, FedProx, LG-FedAvg, FedRep, our method dFLMoE (RCNN) and dFLMoE (SRRResNet). Our framework can recover more details.

where *Attention* is the attention mechanism function [46]. More details can be found in supplementary materials.

## 4. Experiments

### 4.1. Tasks and Datasets

We verify the effectiveness of dFLMoE on 5 non-IID tasks.

**A. Medical image classification (different resolution).** We use the Breast Cancer Histopathological Image Database (BreakHis) [45]. We perform  $x2\downarrow$ ,  $x4\downarrow$ , and  $x8\downarrow$  downsam-

pling on the high-resolution images [49]. Each resolution of medical images is treated as a client, resulting in four clients in total. The dataset for each client was randomly divided into training and testing sets at a ratio of 7:3, following previous work. For the same image with different resolutions, they will be used in either the training set or the testing set. For the model homogeneous framework, we employed ResNet{5}. For the model heterogeneous framework, we employed ResNet{17, 11, 8, 5}.

**B. Medical image super-resolution.** We use BreakHis

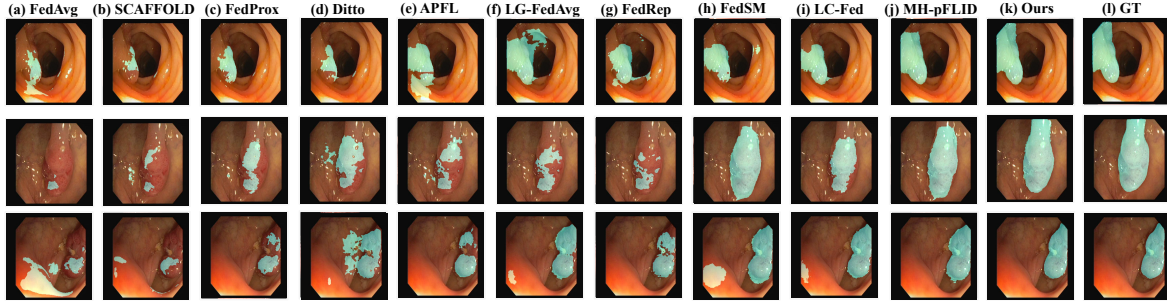


Figure 5. Visualized comparison of Federated Learning in medical image segmentation. We randomly select three samples from different clients to form the visualization. (a-k) Segmentation results for FedAVG, SCAFFOLD, FedProx, Ditto, APFL, LG-FedAvg, FedRep, FedSM, LC-Fed, MH-FLID and our method dFLMoE; (l) Ground truths (denoted as ‘GT’).

Table 2. The results of super-resolution with homogeneous models or heterogeneous models. The  $x8\uparrow$ ,  $x4\uparrow$ , and  $x2\uparrow$  are two times, four times, and eight times super-resolution for downsampling eighth, quarter, and half of high-resolution images. We evaluate PSNR and SSIM results on the BreakHis dataset. The larger the better. The **red** boxes represent the method of individual clients adopting the homogeneous model settings (RCNN). The **blue** boxes represent the method of using heterogeneous models. The three client models are set to SRResNet{18, 12, 6}, respectively. In two different model settings, dFLMoE achieves the best performance.

Method	$x8\uparrow$		$x4\uparrow$		$x2\uparrow$		Average	
	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	SSIM $\uparrow$
Bicubic	20.75	0.4394	23.21	0.6305	26.60	0.9151	23.52	0.6617
Only Local Training	21.12	0.4872	24.04	0.6634	28.36	0.8631	24.51	0.6712
FedAvg	22.00	0.6572	24.65	0.6802	26.46	0.8188	24.37	0.7187
SCAFFOLD	21.33	0.5633	24.47	0.6817	28.61	0.8398	24.80	0.6949
FedProx	21.77	0.6254	23.92	0.6791	27.60	0.8274	24.43	0.7106
LG-FedAvg	21.50	0.4461	23.63	0.6789	27.02	0.8352	24.05	0.6534
FedRep	22.01	0.6170	24.73	0.6999	29.72	0.8964	25.49	0.7378
Ours	<b>23.43</b>	<b>0.6671</b>	<b>27.59</b>	<b>0.8272</b>	<b>34.82</b>	<b>0.9605</b>	<b>28.61</b>	<b>0.8183</b>
Only Local Training	21.76	0.5141	25.23	0.7423	29.31	0.9022	25.43	0.7195
Ours	<b>23.94</b>	<b>0.6929</b>	<b>28.08</b>	<b>0.8436</b>	<b>35.87</b>	<b>0.9686</b>	<b>29.30</b>	<b>0.8350</b>

dataset [45]. We perform  $x2\downarrow$ ,  $x4\downarrow$ , and  $x8\downarrow$  downsampling on the high-resolution images [49]. Each downsampled resolution of medical images is treated as a client, resulting in three clients in total. We used the RCNN [12] for the model heterogeneous framework. We used SRResNet{6, 12, 18} [22] for the model heterogeneous framework.

**C. Medical time-series classification.** We used the Sleep-EDF dataset [13] for the time-series classification task of three clients under non-IID distribution. For the model homogeneous framework, we employed TCN. For the model heterogeneous framework, three clients use the TCN [3], Transformer [57] and RNN [50].

**D. Medical image classification (different label distributions).** This task includes a breast cancer classification task and an ocular disease recognition task. Similar to previous work [52], we also designed eight clients, each using a different model. They are ResNet [15], ShuffleNetV2 [32], ResNeXt [54], SqueezeNet [19], SENet [16], MobileNetV2 [42], DenseNet [17], and VGG [44]. We apply the same non-IID label distribution method as before to the BreakHis

Table 3. The results of time-series classification with homogeneous models or heterogeneous models. We evaluate ACC and MF1 results on the Sleep-EDF dataset. The **red** boxes represent the method of individual clients adopting the homogeneous model settings (TCN). The **blue** boxes represent the method of using heterogeneous models. The three client models are TCN, Transformer, and RNN, respectively. In two different model settings, dFLMoE achieves the best performance.

Method	Client 1		Client 2		Client 3		Average	
	ACC $\uparrow$	MF1 $\uparrow$	ACC $\uparrow$	MF1 $\uparrow$	ACC $\uparrow$	MF1 $\uparrow$	ACC $\uparrow$	MF1 $\uparrow$
Only Local Training	0.9073	0.8757	0.8012	0.7933	0.7791	0.7289	0.8292	0.7993
FedAvg	0.8357	0.7281	0.7719	0.7726	0.7418	0.6083	0.7831	0.7030
SCAFFOLD	0.8792	0.8176	0.8473	0.8494	0.7575	0.6242	0.8280	0.7637
FedProx	0.8541	0.7668	0.8154	0.8162	0.7804	0.7179	0.8166	0.7670
FedRep	0.8934	0.8633	0.8367	0.8221	0.7782	0.7341	0.8361	0.8065
LG-FedAvg	0.8797	0.7613	0.8532	0.8568	0.7656	0.6954	0.8328	0.7712
MH-pFLID	0.9392	0.9117	0.8463	0.8321	0.8244	0.7973	0.8700	0.8470
dFLMoE(Ours)	<b>0.9470</b>	<b>0.9303</b>	<b>0.9201</b>	<b>0.9210</b>	<b>0.8451</b>	<b>0.8123</b>	<b>0.9041</b>	<b>0.8879</b>
Only Local Training	0.9073	0.8757	0.8053	0.8001	0.8012	0.7263	0.8379	0.8007
FedMD	0.9334	0.9225	0.7934	0.7966	0.793	0.7072	0.8399	0.8088
FedDF	0.9146	0.8893	0.7988	0.8042	0.7881	0.6855	0.8338	0.7930
pFedDF	0.9173	0.8957	0.827	0.8309	0.8137	0.7713	0.8527	0.8326
DS-pFL	0.9133	0.9033	0.8253	0.8301	0.8042	0.7539	0.8476	0.8291
KT-pFL	0.924	0.9089	0.8419	0.8466	0.8204	0.7722	0.8621	0.8426
MH-pFLID	0.9439	0.9248	0.8725	0.876	0.824	0.7773	0.8801	0.8594
dFLMoE(Ours)	<b>0.9484</b>	<b>0.9319</b>	<b>0.9308</b>	<b>0.9319</b>	<b>0.8617</b>	<b>0.8319</b>	<b>0.9136</b>	<b>0.8986</b>

and ODIR-5K datasets [4] across 8 clients. Specifically, the data distribution varies among clients.

**E. Medical image segmentation.** Here, we focus on polyp segmentation [11]. The dataset consists of endoscopic images collected and annotated from four centers, with each center’s dataset treats as a separate client. We employed Unet [39] for the model homogeneous framework. For the model heterogeneous framework, each clients adopted Unet++ [60], Unet [39], Res-Unet [10], FCN [31]. , respectively.

## 4.2. Results

**Medical image classification (different resolutions).** In this task, we compare dFLMoE with the baseline framework in two different model settings. For the model homogeneous framework, all frameworks use ResNet5. For the model heterogeneous framework, we use the ResNet family. As in previous work, we use the ResNet family for the model heterogeneous framework. Clients with low-resolution images

Table 4. The results of Image Classification Task with Different Label Distributions. This task includes breast cancer classification and Ocular disease recognition. We evaluate ACC and MF1 results in this task. The larger the better. **Bold** number means the best. dFLMoE has the best performance.

Breast Cancer Classification																		
Method	ResNet		shufflenetv2		ResNeXt		squeezeNet		SENet		MobileNet		DenseNet		VGG		Average	
	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑
Only Local Training	0.59	0.455	0.845	0.8412	0.665	0.5519	0.84	0.7919	0.875	0.849	0.755	0.5752	0.855	0.6884	0.875	0.8515	0.7875	0.7005
FedMD	0.692	0.5721	0.823	0.8027	0.704	0.6087	0.875	0.8544	0.907	0.8745	0.762	0.6627	0.835	0.6493	0.842	0.8001	0.8050	0.7281
FedDF	0.721	0.5949	0.817	0.8094	0.723	0.6221	0.893	0.8735	0.935	0.9021	0.757	0.6609	0.847	0.6819	0.833	0.7826	0.8158	0.7409
pFedDF	0.755	0.6536	0.853	0.8256	0.741	0.6237	0.894	0.8742	0.935	0.9021	0.796	0.7219	0.879	0.7095	0.874	0.8521	0.8409	0.7703
DS-pFL	0.715	0.6099	0.792	0.7734	0.765	0.6547	0.899	0.8792	0.935	0.9021	0.794	0.7331	0.853	0.6691	0.851	0.8266	0.8255	0.7560
KT-pFL	0.765	0.6733	0.87	0.8331	0.755	0.6432	0.885	0.8621	0.935	0.9021	0.78	0.6931	0.865	0.6819	0.905	0.9023	0.8450	0.7739
MH-pFLID	0.805	0.6427	0.945	0.9394	0.82	0.7604	<b>0.963</b>	<b>0.9457</b>	<b>0.975</b>	<b>0.9709</b>	<b>0.815</b>	<b>0.7755</b>	0.895	0.7287	<b>0.995</b>	0.9583	0.9016	0.8402
pFLMoE(Ours)	<b>0.875</b>	<b>0.8745</b>	<b>0.975</b>	<b>0.9749</b>	<b>0.825</b>	<b>0.7951</b>	0.945	0.8934	0.965	0.9458	0.805	0.7428	<b>0.945</b>	<b>0.8611</b>	<b>0.995</b>	<b>0.9936</b>	<b>0.9163</b>	<b>0.8852</b>

Ocular Disease Recognition																		
Method	ResNet		shufflenetv2		ResNeXt		squeezeNet		SENet		MobileNet		DenseNet		VGG		Average	
	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑	ACC↑	MF1↑
Only Local Training	0.6813	0.5607	0.6438	0.6406	0.5063	0.5019	0.5625	0.3705	0.8562	0.8532	0.5813	0.4711	0.5563	0.5061	0.8938	0.7273	0.6602	0.5789
FedMD	0.5375	0.2945	0.7375	0.7065	0.475	0.4017	0.5375	0.1748	0.5375	0.4558	0.6188	0.4245	0.6438	0.3916	0.8562	0.6114	0.6180	0.4326
FedDF	0.6938	0.6413	0.7688	0.7609	0.5437	0.5397	0.5688	0.1813	0.6313	0.6288	0.5375	0.5128	0.5563	0.5312	0.8938	0.5254	0.6493	0.5402
pFedDF	0.7312	0.641	0.7438	0.7324	0.6062	0.5443	0.5437	0.4536	0.6562	0.4611	0.5875	0.5095	0.5437	0.518	0.9062	0.7708	0.6648	0.5788
DS-pFL	0.7563	0.6567	0.7625	0.739	0.575	0.5652	0.5813	0.3874	0.8625	0.8625	0.5875	0.5299	0.5875	0.5394	0.8688	0.6018	0.6977	0.6102
KT-pFL	0.7625	0.7144	0.775	0.7566	0.5125	0.4182	0.5688	0.3877	0.85	0.8498	0.6062	0.5078	0.625	0.4726	0.9187	0.8014	0.7023	0.6136
MH-pFLID	0.775	0.6899	0.8188	0.8126	0.635	0.5652	0.5625	<b>0.4487</b>	0.9125	0.9114	0.6125	0.5044	0.6188	0.5756	0.9125	0.8155	0.7310	0.6654
dFLMoE(Ours)	<b>0.8052</b>	<b>0.7354</b>	<b>0.8313</b>	<b>0.8277</b>	<b>0.6562</b>	<b>0.6552</b>	<b>0.6313</b>	0.4333	<b>0.9625</b>	<b>0.9625</b>	<b>0.6313</b>	<b>0.5202</b>	<b>0.6500</b>	<b>0.5833</b>	<b>0.9500</b>	<b>0.8529</b>	<b>0.7647</b>	<b>0.6962</b>

Table 5. For the medical image segmentation task, we evaluate the Dice result on Polyp dataset. The larger the better. **Bold** number means the best. The **red** boxes represent the method of using homogeneous models. Their clients use the Unet. The **blue** boxes represent the method of using heterogeneous models in each client. The four client models are set to Unet++, Unet, ResUnet, and FCN, respectively. dFLMoE achieves the best segmentation results.

Method	Client1	Client2	Client3	Client4	Average
FedAvg	0.5249	0.4205	0.5676	0.5500	0.5158
SCAFFOLD	0.5244	0.3591	0.5935	0.5713	0.5121
FedProx	0.5529	0.4674	0.5403	0.6301	0.5477
Ditto	0.5720	0.4644	0.6648	0.6416	0.5857
APFL	0.6120	0.5095	0.6333	0.5892	0.5860
LG-FedAvg	0.6053	0.5062	0.7371	0.5596	0.6021
FedRep	0.5809	0.3106	0.7088	0.7023	0.5757
FedSM	0.6894	0.6278	0.8021	0.7391	0.7146
LC-Fed	0.6233	0.4982	0.8217	<b>0.7654</b>	0.6772
dFLMoE (Ours)	<b>0.7918</b>	<b>0.6882</b>	<b>0.8808</b>	0.7644	<b>0.7813</b>
Only Local Training	0.7049	0.4906	0.8079	0.7555	0.6897
MH-pFLID	0.7565	0.6830	0.8644	0.7644	0.7671
dFLMoE (Ours)	<b>0.7945</b>	<b>0.6859</b>	<b>0.8709</b>	<b>0.7710</b>	<b>0.7806</b>

employ shallower models, while clients with high-resolution images use more complex models. In Tab. 1, experimental results show that in both model settings, dFLMoE achieves the best performance. This indicates that dFLMoE can effectively integrate knowledge from both homogeneous or heterogeneous models, thereby enhancing the performance of local models.

**Medical image super-resolution.** This task involves reconstructing different low-resolution medical images into high-resolution images. We consider all images of the same resolution as a single client. In this task, we use the RCNN for the model homogeneous framework and the SResNet family for the model homogeneous framework. As shown in Tab. 2, dFLMoE achieves the best results. Moreover, as shown in Figure 4, our framework can recover more details.

**Time-series classification.** The experimental results in Tab. 3 show that dFLMoE achieves the best results under two

different model settings. This further demonstrates the superiority of dFLMoE in federated learning of homogeneous and heterogeneous models.

**Medical image classification (different label distributions).** In Tab. 4, the experimental results for the medical image classification task with different label distributions, where each client uses heterogeneous models, show that dFLMoE achieves the optimal results. This demonstrates that, compared to heterogeneous federated learning methods, the Mixture of Experts approach of dFLMoE can more effectively fuse knowledge from other clients to make decisions.

**Medical image segmentation.** We validate the effectiveness of dFLMoE in medical image segmentation tasks. Tab. 5 presents the results of federated learning in the segmentation task, demonstrating that dFLMoE achieves the best experimental outcomes under two different model settings. The experimental results not only demonstrate that dFLMoE effectively enhances local model performance, but also prove its applicability to various medical tasks. Meanwhile, the visualization results in Figure 5 show that the segmentation results of dFLMoE are closer to ground truth.

**Connection robustness.** As shown in Tab. 7, We design two disconnect experiments for medical image classification (different resolutions) and medical image segmentation tasks to verify that dFLMoE can still help improve local model training performance in disconnect scenarios. Communication disconnect refers to randomly dropping clients' upload or download processes. Client disconnect means that the corresponding client does not participate in the federated learning. The experimental results are shown in Tab. 6. In the communication disconnect experiment, the results show that compared to centralized solutions, our method experiences lower performance degradation as the dropout rate increases. When the disconnect rate reaches 75%, the cen-

Table 6. The disconnect experiment of dFLMoE and MH-pFLID (centralized Federated Learning) in medical image classification (different resolutions) and medical image segmentation tasks. In the communication disconnect, we randomly disconnect each client’s upload or download operations with the server. At a disconnect rate of 50%, centralized federated learning ensures that each client maintains at least one upload or download operation. At a dropout rate of 75%, it becomes only local training. In the client disconnect, we directly remove certain clients during the federated learning process. For example, a disconnect rate of 25% indicates that only three clients participate in the federated learning, while “None (3 clients)” refers to the performance of three clients out of four. dFLMoE shows less performance degradation compared to the centralized approach in disconnect scenarios.

Communication disconnect					
Task	Method	None	25%	50%	75%
		ACC	ACC	ACC	ACC
Classification	dFLMoE (Ours)	0.8880	0.8798	0.8474	0.8011
	MH-pFLID	0.8583	0.8393	0.7687	0.7258
Segmentation	dFLMoE (Ours)	Dice	Dice	Dice	Dice
	MH-pFLID	0.7860	0.7789	0.7423	0.7211
		0.7671	0.7641	0.7043	0.6897
Client disconnect					
Task	Method	None	25%	None	50%
		(3 client)	ACC	(2 client)	ACC
Classification	dFLMoE (Ours)	0.8771	0.8633	0.8638	0.8474
	MH-pFLID	0.8447	0.8193	0.8340	0.8087
Segmentation	dFLMoE (Ours)	Dice	Dice	Dice	Dice
	MH-pFLID	0.7873	0.7642	0.7838	0.7446
		0.7681	0.7359	0.7737	0.7154

Table 7. Difference between communication disconnect and client disconnect in centralized and decentralized federated learning.

	Centralized	Decentralized
Communication disconnect	Randomly disconnect each client’s upload or download operations with the central server.	Randomly disconnect the upload or download operations between each client.
Client disconnect	Remove the corresponding clients during the federated learning process.	

Table 8. In heterogeneous model settings, we compare the impact of expert parameter quantities on performance in medical image segmentation, time-series classification, breast cancer classification (with different label distributions), and medical super-resolution tasks. #Params represents the average amount of parameters a client needs to share in one communication. The experimental results show that using the entire local model as experts leads to limited performance improvement.

Expert	Segmentation		Time-series	
	#Params(M)	Dice	#Params(M)	ACC
Head	0.001	0.7806	0.002	0.9136
Entire local model	24.015	0.7921	1.181	0.9122
Expert	Breast Cancer		Super-resolution	
	#Params(M)	ACC	#Params(M)	PSNR
Head	0.004	0.9163	0.001	29.30
Entire local model	9.763	0.9077	7.321	29.43

tralized solution performs similarly to only local training, while our approach allows for knowledge transfer, thereby enhancing local model performance. In the client disconnect experiment, dFLMoE still shows less performance degrada-

Table 9. The ablation experiments of dFLMoE. We remove some essential modules to verify the effectiveness of each module. We perform experiments on Time-series classification, medical image super-resolution, and segmentation tasks. We observe that though those experiments outperform centralized methods, they suffer different levels of performance decrease. (MoE: Mixture of Experts; FST: Feature space transform)

Methods	Time-series		Super-resolution		Segmentation
	ACC↑	MF1↑	PSNR↑	SSIM↑	Dice↑
dFLMoE (Ours)	<b>0.9041</b>	<b>0.8879</b>	<b>29.30</b>	<b>0.8350</b>	<b>0.7860</b>
w/o MoE module	0.8731	0.8533	28.65	0.8234	0.7344
w/o FST module	0.8812	0.8681	28.44	0.8261	0.7421
w/ centralized MoE& FST	0.8609	0.8347	27.46	0.8199	0.6625
w/ aggregated head	0.8361	0.8065	26.07	0.7891	0.5893

tion compared to the centralized approach.

**Experts number of parameters.** As shown in Tab. 8, under heterogeneous model settings, we compare the impact of expert parameter quantity on performance in four tasks. The experimental results show that using the entire model as an expert leads to limited performance improvements, but significantly increases the average parameter quantity that each client needs to share, resulting in a higher communication burden.

**Ablation studies.** To verify the effectiveness of the proposed components in dFLMoE, a comparison between dFLMoE and its four components on time-series classification, super-resolution, and segmentation tasks is given in Tab. 9. The four components are as follows: (1) w/o MoE: we replace our designed MoE with the original MoE. (2) w/o FST indicates that we delete the feature space transform module in the local network. (3) w/ centralized MoE& FST or w/ aggregated head means that all clients’ MoE and FST or head parameters are uploaded to the central server for aggregation. Experimental results show that our designed MoE and FT modules effectively integrate knowledge from various clients. Compared to centralized aggregation, our decentralized approach better utilizes knowledge from other clients to enhance local model performance.

## 5. Conclusions

Centralized Federated Learning could lead to knowledge damage during aggregation, and the knowledge would be undermined before it reaches back to each client. It also creates a dependency on the central server, which may affect training stability if the server malfunctions or connections are unstable. We design a decentralized federated learning framework named dFLMoE to address the issues of centralized Federated Learning. dFLMoE shares each client’s head model as an expert with other clients and uses the MoE approach to fuse the knowledge from these experts to make the final decision. We demonstrate the effectiveness of our framework in 5 Non-IID medical tasks under two model settings and achieves state-of-the-art performance.

## References

- [1] Vidushi Agarwal, Shruti Mishra, and Sujata Pal. Towards a sustainable blockchain: A peer-to-peer federated learning based approach. *ACM Transactions on Internet Technology*, 2024. 3
- [2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. 2
- [3] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 6
- [4] Amit Bhati, Neha Gour, Pritee Khanna, and Aparajita Ojha. Discriminative kernel convolution network for multi-label ophthalmic disease detection on imbalanced fundus image dataset. *Computers in Biology and Medicine*, 153:106519, 2023. 6
- [5] Gal Blecher and Shai Fine. Moeatt: A deep mixture of experts model using attention-based routing gate. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 1018–1024. IEEE, 2023. 4
- [6] Qian Chen, Zilong Wang, Yilin Zhou, Jiawei Chen, Dan Xiao, and Xiaodong Lin. Cf: Cluster federated learning in large-scale peer-to-peer networks. In *International Conference on Information Security*, pages 464–472. Springer, 2022. 3
- [7] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020. 2
- [8] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *ICML*, pages 2089–2099. PMLR, 2021. 1, 2
- [9] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. 1, 2
- [10] Foivos I Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:94–114, 2020. 6
- [11] Bo Dong, Wenhai Wang, Deng-Ping Fan, Jinpeng Li, Huazhu Fu, and Ling Shao. Polyp-pvt: Polyp segmentation with pyramid vision transformers. *arXiv preprint arXiv:2108.06932*, 2021. 6
- [12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014. 6
- [13] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000. 6
- [14] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020. 2
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6
- [16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 6
- [17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6
- [18] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *AAAI*, pages 7865–7873, 2021. 2
- [19] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 6
- [20] Sohei Itahara, Takayuki Nishio, Yusuke Koda, Masahiro Morikura, and Koji Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, 22(1):191–205, 2023. 1, 2
- [21] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, pages 5132–5143. PMLR, 2020. 2
- [22] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 6
- [23] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *CoRR*, abs/1910.03581, 2019. 1, 2
- [24] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020. 2
- [25] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *ICML*, pages 6357–6368. PMLR, 2021. 2
- [26] Zexi Li, Jiaxun Lu, Shuang Luo, Didi Zhu, Yunfeng Shao, Yinchuan Li, Zhimeng Zhang, Yongheng Wang, and Chao Wu. Towards effective clustered federated learning: A peer-to-peer framework with adaptive neighbor matching. *IEEE Transactions on Big Data*, 2022. 3
- [27] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020. 1, 2
- [28] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated

- learning. In *Advances in Neural Information Processing Systems*, pages 2351–2363. Curran Associates, Inc., 2020. [1](#), [2](#)
- [29] Jinhua Liu, Christian Desrosiers, and Yuanfeng Zhou. Attmoe: attention-based mixture of experts for nuclear and cytoplasmic segmentation. *Neurocomputing*, 411:139–148, 2020. [4](#)
- [30] Yuan Liu, Zhengpeng Ai, Shuai Sun, Shuangfeng Zhang, Zelei Liu, and Han Yu. Fedcoin: A peer-to-peer payment system for federated learning. In *Federated learning: privacy and incentive*, pages 125–138. Springer, 2020. [3](#)
- [31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. [6](#)
- [32] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. [6](#)
- [33] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020. [2](#)
- [34] Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In *ICML*, pages 15070–15092. PMLR, 2022. [2](#)
- [35] Othmane Marfoq, Giovanni Neglia, Richard Vidal, and Laetitia Kameni. Personalized federated learning through local memorization. In *Proceedings of the 39th International Conference on Machine Learning*, pages 15070–15092. PMLR, 2022. [2](#)
- [36] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282. PMLR, 2017. [1](#), [2](#)
- [37] Jed Mills, Jia Hu, and Geyong Min. Multi-task federated learning for personalised deep neural networks in edge computing. *IEEE Transactions on Parallel and Distributed Systems*, 33(3):630–641, 2021. [2](#)
- [38] Zhen Qin, Xueqiang Yan, Mengchu Zhou, and Shuiguang Deng. Blockdf: A blockchain-based fully decentralized peer-to-peer federated learning framework. In *Proceedings of the ACM on Web Conference 2024*, pages 2914–2925, 2024. [3](#)
- [39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. [6](#)
- [40] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. Braintorrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731*, 2019. [3](#)
- [41] Jose L Salmeron, Irina Arévalo, and Antonio Ruiz-Celma. Benchmarking federated strategies in peer-to-peer federated learning for biomedical data. *Heliyon*, 9(6), 2023. [3](#)
- [42] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. [6](#)
- [43] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020. [2](#)
- [44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [6](#)
- [45] Fabio A. Spanhol, Luiz S. Oliveira, Caroline Petitjean, and Laurent Heutte. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, 63(7):1455–1462, 2016. [5](#), [6](#)
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. [5](#)
- [47] Han Wang, Luis Muñoz-González, David Eklund, and Shahid Raza. Non-iid data re-balancing at iot edge with peer-to-peer federated learning for anomaly detection. In *Proceedings of the 14th ACM conference on security and privacy in wireless and mobile networks*, pages 153–163, 2021. [3](#)
- [48] Jiacheng Wang, Yueming Jin, and Liansheng Wang. Personalizing federated medical image segmentation via local calibration. In *ECCV*, pages 456–472. Springer, 2022. [2](#)
- [49] Luyuan Xie, Cong Li, Zirui Wang, Xin Zhang, Boyan Chen, Qingni Shen, and Zhonghai Wu. Shisrnet: Super-resolution and classification network for low-resolution breast cancer histopathology image, 2023. [5](#), [6](#)
- [50] Luyuan Xie, Cong Li, Xin Zhang, Shengfang Zhai, Yuejian Fang, Qingni Shen, and Zhonghai Wu. Trls: A time series representation learning framework via spectrogram for medical signal processing, 2024. [6](#)
- [51] Luyuan Xie, Manqing Lin, Siyuan Liu, ChenMing Xu, Tianyu Luan, Cong Li, Yuejian Fang, Qingni Shen, and Zhonghai Wu. pffe: Cross-silo personalized federated learning via feature enhancement on medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 599–610. Springer, 2024. [2](#)
- [52] Luyuan Xie, Manqing Lin, Tianyu Luan, Cong Li, Yuejian Fang, Qingni Shen, and Zhonghai Wu. Mh-pflid: Model heterogeneous personalized federated learning via injection and distillation for medical data analysis. *arXiv preprint arXiv:2405.06822*, 2024. [1](#), [2](#), [6](#)
- [53] Luyuan Xie, Manqing Lin, ChenMing Xu, Tianyu Luan, Zhipeng Zeng, Wenjun Qian, Cong Li, Yuejian Fang, Qingni Shen, and Zhonghai Wu. Mh-pflgb: Model heterogeneous personalized federated learning via global bypass for medical image analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 534–545. Springer, 2024. [1](#), [2](#)
- [54] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep

- neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 6
- [55] Liping Yi, Han Yu, Gang Wang, and Xiaoguang Liu. Fedlora: Model-heterogeneous personalized federated learning with lora tuning. *CoRR*, abs/2310.13283, 2023. 2
- [56] Liping Yi, Han Yu, Gang Wang, and Xiaoguang Liu. pFedes: Model heterogeneous personalized federated learning with feature extractor sharing. *CoRR*, abs/2311.06879, 2023. 1, 2
- [57] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. 2021. 6
- [58] Jie Zhang, Song Guo, Xiaosong Ma, Haozhao Wang, Wenchao Xu, and Feijie Wu. Parameterized knowledge transfer for personalized federated learning. In *Advances in Neural Information Processing Systems*, pages 10092–10104. Curran Associates, Inc., 2021. 1, 2
- [59] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, and Dusit Niyato. Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system. *arXiv preprint arXiv:1906.10893*, pages 2327–4662, 2019. 3
- [60] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE transactions on medical imaging*, 39(6):1856–1867, 2019. 6