

# PM-MOE: Mixture of Experts on Private Model Parameters for Personalized Federated Learning

Yu Feng

Beijing Univ. of Posts and Telecomm.  
Beijing, China  
fydannis@bupt.edu.cn

Zongfu Han

Beijing Univ. of Posts and Telecomm.  
Beijing, China  
michan325@bupt.edu.cn

Haoran Luo

Beijing Univ. of Posts and Telecomm.  
Beijing, China  
luohaoran@bupt.edu.cn

Yangli-ao Geng

Beijing Jiaotong University.  
Beijing, China  
gengyla@bjtu.edu.cn

Xie Yu

Beijing Univ. of Aeronautics and  
Astronautics.  
Beijing, China  
yuxie\_scse@buaa.edu.cn

Mengyang Sun

Tsinghua University  
Beijing, China  
sunny19@mails.tsinghua.edu.cn

Meina Song

Beijing Univ. of Posts and Telecomm.  
Beijing, China  
mnsong@bupt.edu.cn

Yifan Zhu\*

Beijing Univ. of Posts and Telecomm.  
Beijing, China  
yifan\_zhu@bupt.edu.cn

Kaiwen Xue

Beijing Univ. of Posts and Telecomm.  
Beijing, China  
xkw@bupt.edu.cn

Guangwei Zhang

Beijing Univ. of Posts and Telecomm.  
Beijing, China  
gwzhang@bupt.edu.cn

## Abstract

Federated learning (FL) has gained widespread attention for its privacy-preserving and collaborative learning capabilities. Due to significant statistical heterogeneity, traditional FL struggles to generalize a shared model across diverse data domains. Personalized federated learning addresses this issue by dividing the model into a globally shared part and a locally private part, with the local model correcting representation biases introduced by the global model. Nevertheless, locally converged parameters more accurately capture domain-specific knowledge, and current methods overlook the potential benefits of these parameters. To address these limitations, we propose PM-MoE architecture. This architecture integrates a mixture of personalized modules and an energy-based personalized modules denoising, enabling each client to select beneficial personalized parameters from other clients. We applied the PM-MoE architecture to nine recent model-split-based personalized federated learning algorithms, achieving performance improvements with minimal additional training. Extensive experiments on six widely adopted datasets and two heterogeneity settings validate

the effectiveness of our approach. The source code is available at <https://github.com/dannis97500/PM-MOE>.

## CCS Concepts

- Computing methodologies → Distributed computing methodologies.

## Keywords

Personalized Federated Learning; Mixture of Experts; Energy-based denoising

## ACM Reference Format:

Yu Feng, Yangli-ao Geng, Yifan Zhu, Zongfu Han, Xie Yu, Kaiwen Xue, Haoran Luo, Mengyang Sun, Guangwei Zhang, and Meina Song. 2025. PM-MOE: Mixture of Experts on Private Model Parameters for Personalized Federated Learning. In *Proceedings of the ACM Web Conference 2025 (WWW '25), April 28–May 2, 2025, Sydney, NSW, Australia*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3696410.3714561>

## 1 Introduction

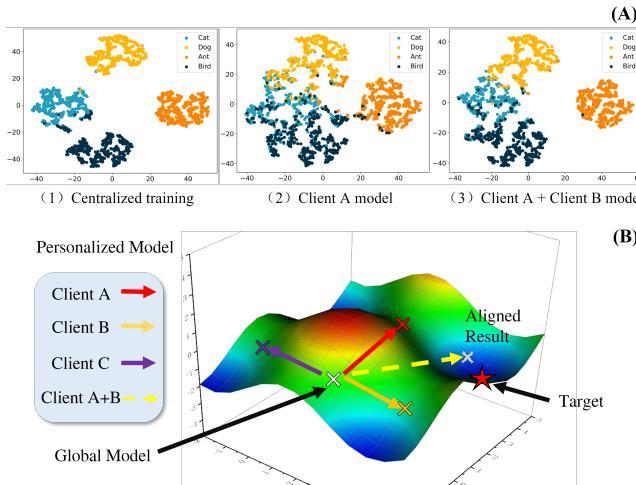
The success of modern methods [9, 15, 24, 43, 58] is largely driven by the growing availability of training data [18, 26, 27, 48]. Unfortunately, there are still vast amounts of isolated data remain underutilized due to strict privacy requirements [10, 44]. As a result, federated learning (FL) [1, 5, 17, 37, 41, 55], has gained significant attention for its strong privacy protection and collaborative learning capabilities. This innovative paradigm allows multiple clients to collaboratively train models, where the server only aggregates models and keep private data remaining on each client. Despite its effectiveness, traditional FL methods suffer from performance degradation due to statistical heterogeneity [22]—data domains

\*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

WWW '25, April 28–May 2, 2025, Sydney, NSW, Australia.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-1274-6/25/04  
<https://doi.org/10.1145/3696410.3714561>



**Figure 1: Motivation of our study.** (A) t-SNE graph shows the inference effects of different models on the same set of data. (B) Client A gets closer to the target when using Client B's personalized model, but moves farther from the target when using Client C's personalized model.

on each client are biased, with uneven class distributions, varying sample sizes, and significant feature differences.

Personalized federated learning (PFL) [32, 33, 39, 47] alleviates this limitation by allowing each client to better fit local data. Specifically, PFL methods focus on balancing local personalization with global consistency by splitting models into global and personalized modules [42, 62], where personalized modules capture unique local data characteristics, mitigating global model biases and better adapting to individual client data. Recent efforts have been developed based on meta-learning [13], regularization [11, 31], model splitting [4], knowledge distillation [45–47, 52, 54], and personalized aggregation [33, 39, 61].

Given that the same types of data can be distributed across multiple clients, then a key question arises: “**Can personalized modules from different clients mutually enhance each other’s performance?**”, which is overlooked in current PFL methods. To investigate, we conducted experiments based on the state-of-the-art PFL approaches. We randomly select a client, and several data categories which distributed across different clients. Subsequently, we trained in both centralized and personalized federated learning manner. By comparison, We evaluated whether integrating personalized parameters from other clients could improve model’s representation capability. As illustrated in Figure 1 (A), the selected client indeed benefited from the personalized modules from another client.

Driven by the above analysis, in this paper, we aim to explore how personalized modules from different clients can mutually enhance performance. As illustrated in Figure 1 (B), all clients utilize the same global model, while applying personalized module to debias according to the local data domain. For a single client, not all personalized modules contribute positively to the final representation. Therefore, we leverage two basic principles when designing our

model: 1) Dynamically weighting the effect of personalized modules based on the current input. 2) Filtering modules that exhibit negative effects;

In this paper, we introduce PM-MoE, a two-stage personalized federated learning framework based on mixture of experts (MoE) architecture [40, 49]. In the first stage, we pretrained models to get global and personalized modules; In the second stage, we proposed the mixture of personalized modules method (MPM) and the energy-based denoising method(EDM) to make the personalized modules from different clients enhance each other. **With the first principle**, PM-MoE employs the MPM based on MoE gate selection. **With the second principle**, PM-MoE incorporates the EDM to filter out noisy personalized models. Together, these two components enable personalized modules from different clients to mutually enhance each other. Additionally, sharing converged personalized parameters will not break privacy requirements due to there is no gradient leakage during training. We evaluated PM-MoE on nine SOTA PFL benchmarks across six popular federated learning datasets. The experimental results demonstrate PM-MoE consistently improves the performance of various PFL methods.

In summary, we conclude our contributions as follows:

- We propose PM-MOE, a novel two-stage framework for personalized federated learning which exchanges personalized knowledge across clients. In the first stage, the PM-MOE pretrains PFL models, followed by a fine-tuning stage for knowledge exchanges.
- Specifically, PM-MOE employs a simple MOE structure to dynamically weighting the contribution of different personalized modules. Besides, PM-MOE introduces an energy-based denosing method to filter those clients with negative effects.
- We conduct extensive experiments to nine state-of-art PFL methods across six datasets. The experimental results demonstrate PM-MOE’s consistently improvement on various settings.

## 2 Notations and Preliminaries

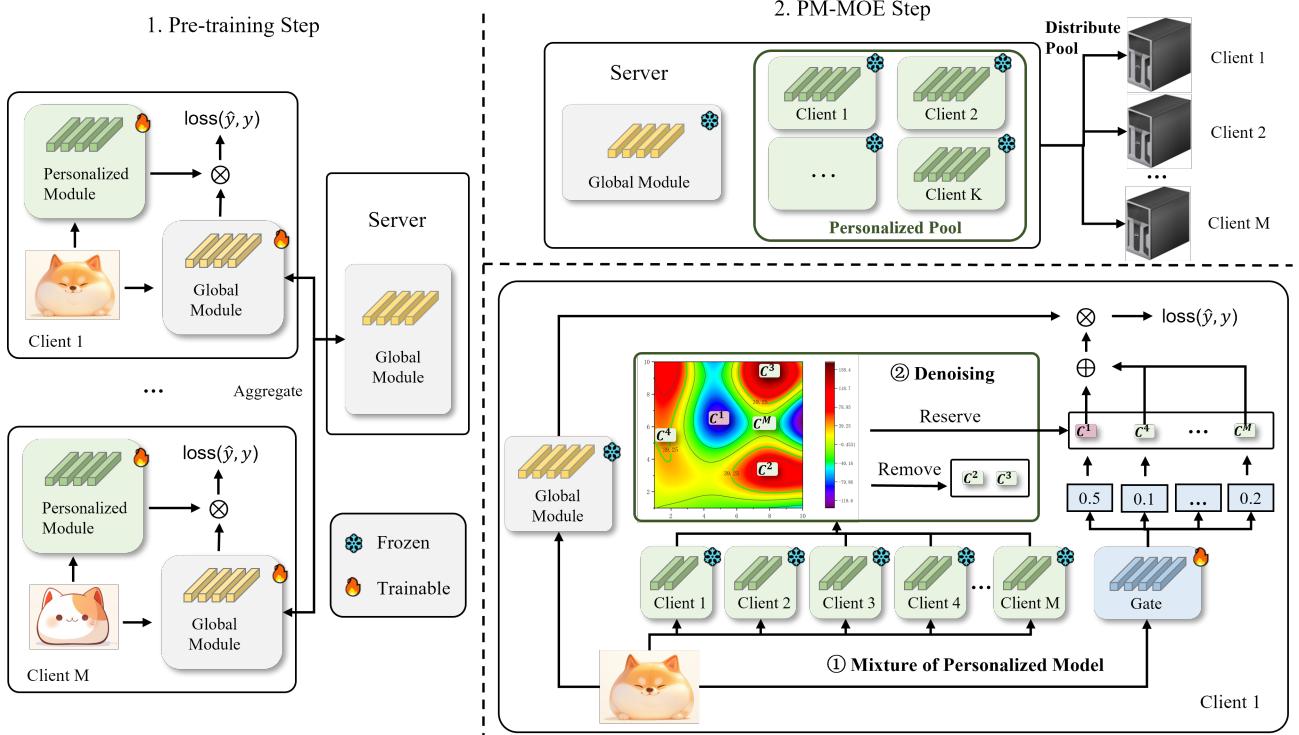
### 2.1 Notations

In PFL,  $M$  clients share the same model structure. Here, we denote notations following FedGen [50] and FedRep [19]. Each client is denoted as  $C^j$  ( $j \in 1, 2, \dots, M$ ), having its own data domain  $\mathcal{D}^j$  with  $N^j$  samples ( $j \in 1, 2, \dots, M$ ). The data distribution of  $\mathcal{D}^j$  is denoted as  $P^j$ . Specifically,  $\mathcal{D}^j = \{x_i^j, y_i^j\}_{i=1}^{N^j}$ , where  $i$  is the number of training samples.  $x_i$  is the  $i$ -th data sample and  $y_i$  is its corresponding label. Each client  $C^j$  in PFL has two modules: the global module and the personalized module, which is denoted as  $W_g^j$  and  $W_p^j$  respectively.

### 2.2 Preliminaries

In a typical PFL method, there is a centralized server who firstly aggregates clients’ global modules  $\{W_g^1, W_g^2, \dots, W_g^M\}$ , and then distributes the aggregated module  $W_g$  to each client. Therefore, each client is required to firstly train on  $\mathcal{D}^j$  and upload their  $W_g^j$  every  $E_l$  iteration. The sever aggregates global modules by the function  $f$  as:

$$W_g = \frac{1}{N} \sum_{j=1}^M N^j f(W_g^j), \quad (1)$$



**Figure 2: Overall Architecture of Personalized Model parameters with Mixture of Experts**

where  $N = \sum_{j=1}^M N^j$  and  $f$  can be algorithms like FedAvg [41], FedProx [38], etc. After aggregation, the server sends  $W_g$  to client  $C^j$ . Then, client  $C^j$  enters the next training. Therefore, the objective loss function  $\mathcal{L}$  for the entire personalized federated learning task is as follows:

$$\min_{W_g^j, W_p^j} \mathcal{L} = \min \sum_{j=1}^M \mathbb{E}_{(x^j, y^j) \sim P_j} [L^j(x^j, y^j; W_g^j, W_p^j)]. \quad (2)$$

Here,  $L^j$  is the loss function for client  $C^j$ .

### 3 Method

#### 3.1 The PM-MOE Overall Framework

In this section, we introduce the overall framework of PM-MOE, which is a two-stage training framework. Specifically, our contributions lie in the mixture of personalized modules (MPM) and an energy-based denoising method (EDM). The MPM addresses the challenge of effectively utilizing personalized models, while the EDM method removes those personalized models with negative effects.

The training process of PM-MOE is divided into two steps, as shown in Figure 2. In pre-training step, we train model and obtain its converged global and personalized modules for each client, thereby constructing a personalized prompt pool. In PM-MOE step, we leverage the proposed MPM and EDM to select the optimal combination among personalized modules for each client. The following sections provide a detailed explanation of these two key phases.

*Phase 1: Pre-training.* The statistically heterogeneous distribution data  $\mathcal{D}^j$  of client  $C^j$  is mapped to the feature space  $x_{g,rep}^j$  through the global feature extractor  $f_g : \mathbb{R}^U \rightarrow \mathbb{R}^D$ , and to the feature space  $x_{p,rep}^j$  via the personalized feature extractor  $f_p : \mathbb{R}^U \rightarrow \mathbb{R}^D$ . The weighted aggregated feature space  $x_{rep}^j = x_{g,rep}^j + x_{p,rep}^j$  is then mapped to the corresponding label space through the global classifier  $s_g : \mathbb{R}^D \rightarrow \mathbb{R}^C$  and the personalized classifier  $s_p : \mathbb{R}^D \rightarrow \mathbb{R}^C$ .  $U, D$  and  $C$  represent the input space, feature space, and label space, respectively.

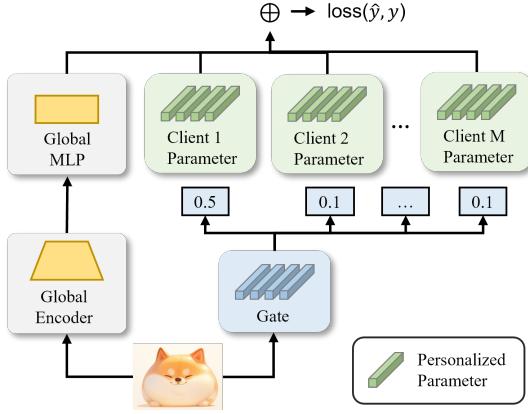
$$x_{rep}^j = f_g(W_{g,fe}^j, x^j) + f_p(W_{p,fe}^j, x^j). \quad (3)$$

Additionally, as seen in DBE [59], there exists a personalized vector parameter  $PP^j \in \mathbb{R}^D$  to correct the local data distribution. The associated expressions are as follows:

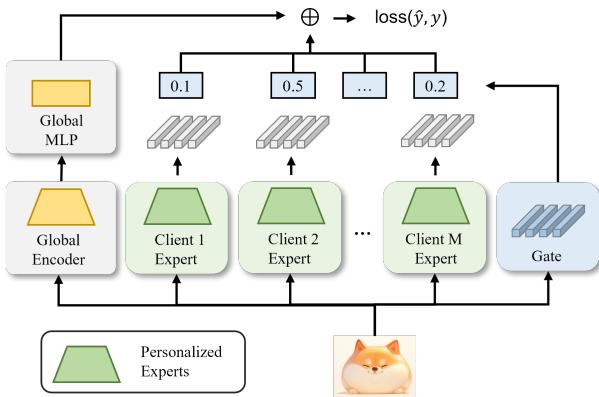
$$\hat{y}^j = s_g(W_{g,hd}^j, h^j) + s_p(W_{p,hd}^j, h^j) + PP^j. \quad (4)$$

During training, the global model parameters  $W_{g,fe}^j$  and  $W_{g,hd}^j$  are uploaded to the server for aggregation, while the personalized model parameters  $W_{p,fe}^j$ ,  $W_{p,hd}^j$  and  $PP^j$  are computed locally and not uploaded. After the global training process with  $E_g$  epochs, the model converges.

*Phase 2: PM-MOE Fine-Tuning.* First, after the convergence of the model-splitting-based series of models, the server collects the trained personalized model parameters to form a personalized parameter pool, which is then distributed to each client. Next, each client locally trains a gating network, which assigns weights to



**Figure 3: Diagram of Mixture of Personalized Parameters.**



**Figure 4: Diagram of Mixture of Personalized Experts.**

each personalized model based on the input data, thereby effectively utilizing the personalized knowledge from all clients. For detailed information, refer to Section 3.2. Finally, since some of the personalized knowledge from other clients is irrelevant to the local data distribution, training the gating network with these noisy parameters can hinder convergence. To address this, we designed an energy-based denoising method. For further details, see Section 3.3.

### 3.2 Mixture of Personalized Modules

PM-MOE is a flexible architecture, and to accommodate the complex and diverse model-splitting-based personalized federated learning algorithms, we designed two adaptation methods: MPP and MPE, as shown in Figures 3 and 4.

Assume that a personalized federated learning algorithm involves personalized parameters, these parameters do not project data into vectors of other dimensions. We define this type as local personalized parameters ( $PP$ ). Suppose the personalized federated learning algorithm also involves personalized expert models, where the expert models  $\mathcal{W}_p^j$  map data  $\mathcal{D}^j$  to a new feature space. We define this type as local personalized experts ( $PE$ ). The server builds and synchronizes a set of personalized models. Depending on the type of personalized model, the server collects the converged

model parameters from all clients, constructing a personalized parameter pool  $\mathcal{W}_{PP} = \{PP^j\}_{j=1}^M$  and a personalized expert pool  $\mathcal{W}_{PE} = \{W_{PE}^j\}_{j=1}^M$ . The server then synchronizes these sets with all clients.

Clients build a gating network and fine-tune parameters. Since each personalized federated learning client is diverse, as shown in Figure 3, we divide the combination of the gating network and personalized models into two categories. **The first is commonalities.** The calculation of set weights depends on the input data  $x^j$ . To achieve this, we construct gating networks  $G_{PP}^j, G_{PE}^j$  for the personalized parameter and the personalized expert, with corresponding training parameters  $\theta_{PP}^j, \theta_{PE}^j$ . The weight calculations are formally represented as follows:

$$\alpha_{PP} = G_{PP}^j(x^j, \theta_{PP}^j) \quad (5)$$

$$\alpha_{PE} = G_{PE}^j(x^j, \theta_{PE}^j) \quad (6)$$

We then sort the weights calculated by formulas (1) and (2) in descending order. From the set, we select the top  $k$  parameters and construct the personalized parameter and expert subsets as  $\{\alpha_{PP}^l\} = Top(k, \alpha_{PP}), \{\alpha_{PE}^l\} = Top(k, \alpha_{PE})$ . Here,  $l$  denotes the index of the selected clients, where  $l \in [1, M]$ .

**Second is Differences.** Since the personalized parameter pool  $\mathcal{W}_{PP}$  does not process data, we directly compute the weighted sum of the personalized parameters, resulting in a vector with the same shape as the local personalized parameter  $PP^j$  as follows:

$$PP_{moe}^j = \mathcal{W}_{PP}^l \cdot \alpha_{PP}^l \quad (7)$$

In our setting, the weighted vector  $\{PP_{moe}\}^j$  replaces the local personalized parameter  $\{PP\}^j$  on client  $C^j$ . For the personalized expert set  $\mathcal{W}_{PE} = W_{PE}^j, M_{j=1}^M$ , taking the personalized classifier  $s_p$  as an example, each expert maps the data to a new feature space  $h^l$ , where:

$$h^l \in \mathbb{R}^C = s_p^l(x^j, W_{PE}^l) \quad (8)$$

We then compute the mixed weighted personalized parameter vector  $x_{moe}^j$  as:

$$x_{moe}^j = h^l \cdot \alpha_{PE}^l \quad (9)$$

The client  $C^j$  then replaces the output of the local personalized expert  $W_{PE}^j$  with  $x_{moe}^j \in \mathbb{R}^C$ .

Since the converged parameters reflect the local data knowledge that each client has spent significant effort training, during the training process, the personalized parameter and expert sets  $\mathcal{W}_{PP}, \mathcal{W}_{PE}$  are frozen and not optimized together with the gating network.

### 3.3 Energy-based Personalized Modules Denoising

Due to MOE using Top-K to select appropriate experts, this ranking based solely on parameter magnitude lacks confidence and introduces noise to some extent. It leads to the gating network optimizing gradients in the wrong direction. To effectively remove noise from the personalized parameter pool, inspired by energy-based models, we propose an energy-based personalized expert denoising method.

**Table 1: Results of federated and personalized federated learning algorithms on six datasets with heterogeneous data distribution (Dirichlet distribution with  $S = 0$  and  $S = 20$ ). Bold: Best performance.**

Spilt Type	$S = 0$						$S = 20$					
Method	MNIST	FMNIST	Cifar10	Cifar100	TINY	AGNews	MNIST	FMNIST	Cifar10	Cifar100	TINY	AGNews
FedAvg	98.93	88.64	63.68	32.94	17.69	62.40	98.95	90.69	67.74	35.37	19.66	71.68
FedProx	98.93	88.50	63.85	33.07	17.60	65.75	98.98	90.79	67.54	35.42	19.56	72.90
SCAFFOLD	99.12	88.74	64.19	34.71	19.67	78.85	99.18	91.44	70.40	38.54	19.67	78.50
FedGEN	98.98	88.79	64.36	32.72	15.85	63.13	98.98	90.90	67.59	34.52	17.70	71.65
MOON	98.92	88.59	63.87	33.00	17.57	62.21	98.98	90.74	67.53	35.36	17.57	71.76
FedPer	99.49	97.53	89.90	48.27	36.36	93.99	98.62	93.57	76.67	36.28	25.91	88.75
LG-FedAvg	99.28	97.25	89.02	47.03	33.20	94.33	97.85	92.23	73.95	35.90	23.78	87.77
FedRep	99.46	97.58	90.19	49.44	38.09	93.78	98.61	93.77	77.25	36.52	25.77	89.02
FedRoD	99.68	97.60	90.07	51.92	38.90	93.65	99.33	94.06	79.50	42.45	29.07	88.91
FedGH	99.29	97.40	84.50	48.61	25.80	92.58	97.94	92.25	73.79	37.88	21.51	88.19
FedBABU	99.67	97.74	91.38	50.83	34.53	92.87	99.32	94.71	82.17	40.46	25.92	87.58
GPFL	99.49	94.91	77.79	57.41	27.08	90.84	99.49	93.21	72.39	49.01	22.92	82.41
FedCP	99.75	98.31	93.76	69.83	65.97	92.40	99.27	94.45	80.29	41.79	31.93	87.62
DBE	98.17	93.95	89.11	60.33	38.29	93.70	96.97	91.11	79.76	52.30	31.11	89.08
<b>PM-MOE</b>	<b>99.85</b>	<b>98.61</b>	<b>93.95</b>	<b>70.68</b>	<b>66.33</b>	<b>94.76</b>	<b>99.49</b>	<b>94.79</b>	<b>82.21</b>	<b>52.36</b>	<b>32.15</b>	<b>89.16</b>

The core idea is to build an energy function to describe the dependency or similarity between inputs. Simply put, the essence of the method is to calculate energy—high energy corresponds to low similarity, while low energy indicates high similarity. Taking the personalized feature extractor experts as an example, for client  $C^j$ , the personalized expert pool  $\mathcal{W}_{PE} = \{W_{PE}^j\}_{j=1}^M$  uses a projection function  $f_p : \mathbb{R}^U \rightarrow \mathbb{R}^D$  to map data  $x^j$  to  $H = \{h^1, h^2, \dots, h^M\}$ . The vector  $h^j \in \mathbb{R}^D$  from the local client is taken as the scalar for energy. Then, the vectors mapped by other client models are projected into the coordinate system of  $h^j$ . Let's define the energy function for a given input pair  $(h^j, h^k)$  as follows:

$$E^k(h^j, h^k) = -v^k \cdot \mathbb{I} \quad (10)$$

where  $v^k \in \mathbb{R}^D = \frac{h^k \cdot h^j}{||h^k|| \cdot ||h^j||}$ , ( $k \neq j$ ). And each dimension of the vector is denoted by  $\mathbb{I} \in D$ . The projected vector set is defined as  $V = \{v^1, \dots, v^k\}$ , ( $k \in [M], k \neq j$ ). Then, Helmholtz free energy can be expressed as the negative logarithm of the partition function:

$$F_T^k(v^k) = -T \log \sum \exp(-E^k(h^j, h^k)/T) \quad (11)$$

Since we use the local client's vector as a fixed anchor, it is natural to choose the negative Helmholtz free energy as the confidence score for the similarity between  $h^k$  and  $h^j$ .

$$H^k(h^j, h^k) = -F_T^k(v^k) = T \log \sum \exp(-E^k(h^j, h^k)/T) \quad (12)$$

where  $T$  is the temperature parameter. Therefore, we use the confidence score to filter out noise (irrelevant experts). Then, we set a dropout ratio coefficient  $\gamma \in (0, 1)$ . The confidence scores of all personalized feature extraction experts are sorted in ascending order, and the bottom  $\gamma$ -proportion of experts are removed.

Finally, after the two modules of PM-MOE framework, the total objective loss of PM-MOE is as follows:

$$\min_{\theta_{PP}^j, \theta_{PE}^j} \mathcal{L} = \min \sum_j^K \mathbb{E}_{(x^j, y^j)} p_j L^j(x^j, y^j; \theta_{PP}^j, \theta_{PE}^j) \quad (13)$$

### 3.4 Theoretical Analysis

In this section, we demonstrate that **leveraging personalized models converged from other clients is more beneficial for improving the performance of local models**. In simple terms, model-split-based personalized federated learning shares uploaded models to learn from the data distribution of all parties involved. However, the heterogeneous nature of the data creates a tug-of-war, resulting in inefficient knowledge transfer between clients. Interestingly, the private parameters that are not uploaded by clients best capture local knowledge.

Therefore, we propose that utilizing these converged personalized models is necessary to enhance performance, leading to the design of the PM-MOE architecture. In this subsection, we theoretically prove that the PM-MOE architecture converges to a lower bound. Even in extreme cases, where each client's data distribution is entirely different, this architecture does not degrade the performance of local models.

**THEOREM 3.1. (Lower Bound on the Final Accuracy of MPE)**  
Suppose there are  $M (\geq 2)$  client experts predicting independently, each with an average accuracy rate of  $p (> 0)$ . If a trained gate network assigns samples to the client experts such that the ratio of the probability of assigning a sample to a correct expert versus an incorrect expert is  $1 + \alpha$ , where  $\alpha > 0$ . Then, the final accuracy of MPE is bounded from below by:

$$P_{MPE} \geq \frac{(1 + \alpha)p}{1 + \alpha(p + \frac{1-p}{M})} > p = P_{client}. \quad (14)$$

We briefly proof the key steps, and other details are given in the appendix.

**PROOF.** We define the event set:

$$\mathcal{A} := \{s \text{ out of } M \text{ experts are able to predict correctly}\}.$$

$$P(\mathcal{A}) = \binom{M}{s} p^s (1-p)^{M-s}, \quad (15)$$

Under the above condition, if the gated network assigns s client weights, the model is able to still correctly predict the sample. We have:

$$P(\mathcal{B} | \mathcal{A}) = \frac{(1+\alpha)s}{(1+\alpha)s + (M-s)} = \frac{(1+\alpha)s}{M+\alpha s}, \quad (16)$$

And then,  $\mathcal{B} := \{\text{MPE can predict correctly}\}$ . We have:

$$\begin{aligned} P(\mathcal{B}) &= \sum_{\mathcal{A}} P(\mathcal{A})P(\mathcal{B} | \mathcal{A}) \\ &= \mathbb{E} \left[ \frac{(1+\alpha)Mp}{M+\alpha(t+1)} \right] \quad (\text{Let } t = s-1). \end{aligned} \quad (17)$$

Define the function:

$$f(t) = \frac{(1+\alpha)Mp}{M+\alpha(t+1)}. \quad (18)$$

We observe that  $f(t)$  is a convex function. And by Jensen's inequality, we have:

$$\mathbb{E}[f(t)] \geq f(\mathbb{E}[t]). \quad (19)$$

Combining (17) and (19) yields

$$P_{\text{MPE}} = P(\mathcal{B}) \geq f(\mathbb{E}[t]) = \frac{(1+\alpha)p}{1+\alpha(p+\frac{1-p}{M})}. \quad (20)$$

The last expression is strictly increasing with respect to  $\alpha$  when  $\alpha > 0$ , and thus:

$$P_{\text{MPE}} \geq \frac{(1+\alpha)p}{1+\alpha(p+\frac{1-p}{M})} > \frac{(1+0)p}{1+0(p+\frac{1-p}{M})} = p = P_{\text{client}}. \quad (21)$$

□

Theorem 3.1 indicates that the accuracy of the MPE is bounded below by the average accuracy of an individual client. Furthermore, the lower bound  $\frac{(1+\alpha)p}{1+\alpha(p+\frac{1-p}{M})}$  increases monotonically with respect to both  $\alpha$  and  $M$ . In other words, the accuracy of the MPE will be improved as the training of the gate network. Specifically, when the gate network is well trained (i.e.,  $\alpha \gg 0$ ) and  $M$  is large enough, the accuracy of MPE will asymptotically approach 100%.

## 4 Experiment

### 4.1 Baseline Methods.

We referred to the Personal Federated Learning library, PFLlib [63]. Furthermore, we compared general federated learning algorithms such as FedAvg [41], FedProx [38], SCAFFOLD [23], MOON [30], and FedGen [66], alongside recent state-of-the-art personalized federated learning methods, including personalized feature extractors like FedGH [56], LG-FedAvg [34], FedBABU [42], FedCP[62], GPFL [60], FedPer[4], FedRep [8], FedRod [6], and DBE [59] for personalized parameters.

### 4.2 Experimental Results

*Main Results.* Tables 1 show that PM-MOE consistently outperforms other personalized federated learning methods across both partitioning settings in tasks ranging from 4 to 200 classes. Compared to traditional federated learning methods, personalized federated learning better handles data heterogeneity, with a performance

improvement of up to 48.64% over the FedAvg baseline. Interestingly, when data heterogeneity decreases, the overall performance of personalized methods also drops. The PM-MOE framework leverages the personalized models converged from all clients to improve each client's performance. If parameters from other clients are noisy, the local gating network assigns weights to prioritize the local personalized model, protecting its performance. Conversely, if external parameters are useful, the network allocates weights accordingly, enhancing the local model's performance.

*Analysis of Gating Network Parameters.* As the most critical component for each client in this framework is the training of the gating network, this section presents parameter experiments focused on tuning the number of layers, activation functions, and initialization parameters of the gating network. In this subsection, to highlight the differences between methods across different dimensions, we will apply sigmoid normalization to the data in the experimental group.

- **Number of Layers in the Gating Network:** We conducted four sets of experiments on the number of layers in the gating network. 1 layer: (input dimension, number of experts); 2 layers: (input dimension, 128, number of experts); 3 layers: (input dimension, 128, 256, number of experts); 4 layers: (input dimension, 128, 256, 128, number of experts). As shown in Figure 5-(a), increasing the depth of the gating network proves to be effective. The gating network needs to determine the weights for all personalized parameters based on input data, requiring deeper neural networks on the client side to capture data features effectively.

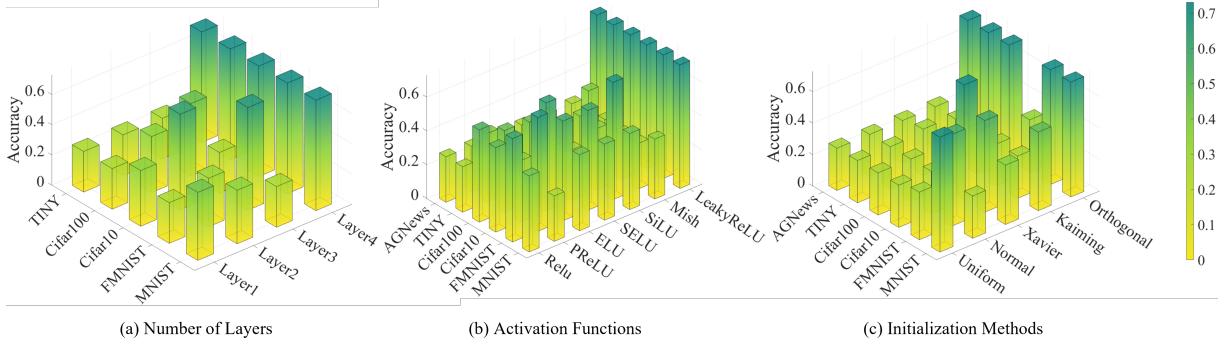
- **Activation Functions of the Gating Network:** For the 4-layer feedforward gating network, we tested common neural network activation functions such as ReLU, LeakyReLU, PReLU, ELU, SELU, SiLU, and Mish. As shown in Figure 5-(b), the most effective activation function is LeakyReLU. LeakyReLU's non-linearity in the negative region allows the neural network to learn and model more complex data, effectively assigning personalized model parameters the appropriate weights.

- **Initialization Methods for the Gating Network:** We compared commonly used parameter initialization methods, including uniform distribution, normal distribution, Xavier, Kaiming, Orthogonal, and Spectral. As shown in Figure 5-(c), most results indicated that the Orthogonal initialization method yields the best performance for gated network models. This method draws initial weights from the orthogonal group, maintaining dynamic isometry throughout the network's learning process, which helps preserve a relatively stable proportional relationship between input and output signals.

*Ablation Study.* In this section, we conducted ablation studies to evaluate the effectiveness of each individually designed module. The experiments confirmed that the PM-MOE component and the denoising component improve the performance of the model-split-based personalized federated learning algorithm. We selected the best performing personalized federated learning methods in the comparison dataset for comparison. As shown in Table 3, adding the MOE component resulted in an average improvement of 0.2% across the 4, 10, and 100 class settings. The regular denoising ratio was

**Table 2: Ablation experiments of PM-MOE across 9 state-of-the-art model-split-based personalized federated learning algorithms.**

Spilt Type	$S = 0$						$S = 20$						Avg $\uparrow$
Method	MNIST	FMNIST	Cifar10	Cifar100	TINY	AGNews	MNIST	FMNIST	Cifar10	Cifar100	TINY	AGNews	-
FedPer	99.49	97.53	89.90	48.27	36.36	93.99	98.62	93.57	76.67	36.28	25.91	88.75	-
+PM-MOE	<b>99.50</b>	<b>97.55</b>	<b>89.96</b>	<b>48.34</b>	<b>36.42</b>	<b>93.99</b>	<b>98.62</b>	<b>93.59</b>	<b>76.69</b>	<b>36.30</b>	<b>25.96</b>	<b>88.75</b>	0.0275
LG-FedAvg	99.28	97.25	89.02	47.03	33.20	94.33	97.85	92.23	73.95	35.90	23.78	87.77	-
+PM-MOE	<b>99.28</b>	<b>97.28</b>	<b>89.19</b>	<b>47.10</b>	<b>33.25</b>	<b>94.76</b>	<b>97.85</b>	<b>92.23</b>	<b>74.01</b>	<b>35.90</b>	<b>23.78</b>	<b>87.77</b>	0.0325
FedRep	99.46	97.58	90.19	49.44	38.09	93.78	98.61	93.77	77.25	36.52	25.77	89.02	-
+PM-MOE	<b>99.47</b>	<b>97.60</b>	<b>90.24</b>	<b>49.49</b>	<b>38.12</b>	<b>93.87</b>	<b>98.61</b>	<b>93.82</b>	<b>77.25</b>	<b>36.52</b>	<b>25.78</b>	<b>89.02</b>	0.0258
FedRoD	99.68	97.60	90.07	51.92	38.90	93.65	99.33	94.06	79.50	42.45	29.07	88.91	-
+PM-MOE	<b>99.69</b>	<b>97.66</b>	<b>90.22</b>	<b>52.82</b>	<b>39.25</b>	<b>93.72</b>	<b>99.33</b>	<b>94.06</b>	<b>79.50</b>	<b>42.60</b>	<b>29.07</b>	<b>88.93</b>	0.1425
FedGH	99.29	97.40	84.50	48.61	25.80	92.58	97.94	92.25	73.79	37.88	21.49	88.10	-
+PM-MOE	<b>99.30</b>	<b>97.40</b>	<b>88.61</b>	<b>48.69</b>	<b>25.82</b>	<b>92.66</b>	<b>97.94</b>	<b>92.25</b>	<b>73.79</b>	<b>37.88</b>	<b>21.51</b>	<b>88.19</b>	0.3675
FedBABU	99.67	97.74	91.38	50.83	34.53	92.87	99.32	94.71	82.17	40.46	25.92	87.58	-
+PM-MOE	<b>99.67</b>	<b>97.76</b>	<b>91.41</b>	<b>50.83</b>	<b>34.53</b>	<b>93.55</b>	<b>99.32</b>	<b>94.79</b>	<b>82.17</b>	<b>40.46</b>	<b>25.92</b>	<b>88.29</b>	0.1267
GPFL	99.49	94.91	77.79	57.41	27.08	90.84	99.49	93.21	72.39	49.01	22.92	82.41	-
+PM-MOE	<b>99.50</b>	<b>95.56</b>	<b>82.28</b>	<b>57.41</b>	<b>27.08</b>	<b>91.94</b>	<b>99.49</b>	<b>93.21</b>	<b>72.39</b>	<b>49.01</b>	<b>22.92</b>	<b>82.41</b>	0.5208
FedCP	99.75	98.31	93.76	69.83	65.97	92.40	99.27	94.45	80.29	41.79	31.93	87.62	-
+PM-MOE	<b>99.85</b>	<b>98.61</b>	<b>93.95</b>	<b>70.68</b>	<b>66.33</b>	<b>92.48</b>	<b>99.30</b>	<b>94.63</b>	<b>80.51</b>	<b>42.49</b>	<b>31.96</b>	<b>87.67</b>	0.2575
DBE	98.17	93.95	89.11	60.33	38.29	93.70	96.97	91.11	79.76	52.30	31.10	89.08	-
+PM-MOE	<b>99.63</b>	<b>97.23</b>	<b>89.90</b>	<b>60.53</b>	<b>38.36</b>	<b>93.74</b>	<b>99.38</b>	<b>93.40</b>	<b>80.05</b>	<b>52.30</b>	<b>31.11</b>	<b>89.08</b>	0.9033

**Figure 5: Results of Gating Network Parameters****Table 3: Ablation Experiment Analysis Results**

Method	AGNews	FMNIST	Cifar100	Avg $\uparrow$
pFL	94.33	98.31	69.83	-
+MOE	94.52	98.39	70.12	0.19%
+MOE+Denoising	94.76	98.61	70.68	0.53%

set to 0.2, and adding the denoising component led to an average improvement of 0.53%.

In detail, we demonstrated that our proposed PM-MOE framework improves nine state-of-the-art personalized federated learning algorithms. Specifically, we used data heterogeneity settings of  $S = 0$  (100% heterogeneity) and  $S = 20$  (80% heterogeneity). As shown in Table 2, PM-MOE enhances the performance of all personalized federated learning algorithms across six widely adopted datasets.

#### Model Parameter Analysis.

- **Top-k Impact Analysis:** The number of personalized parameters determines the breadth of knowledge. If top  $k$  is too small, it may not fully utilize knowledge from other clients. If top  $k$  is too large, it may introduce excessive noisy knowledge. Therefore, we conducted extensive experiments on the choice of  $k$ . Keeping other conditions constant, we set  $k = 2, 4, 8, 16, 20$  across 20 clients. As shown in Figure 6, in highly heterogeneous data settings ( $S = 0$ ), many clients do not share categories. Thus, setting  $k$  to half the number of clients helps the gating network select more effective personalized parameters. In settings with some shared data ( $S = 20$ ), where each client shares a few categories, a larger  $k$ , typically equal to the number of clients, is preferable as it allows the gating network to reference personalized knowledge from all clients.

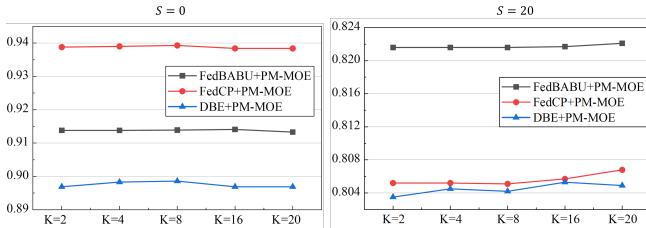


Figure 6: Impact of Top k in PM-MOE.

- Gating Network Learning Rate Analysis:** Keeping other variables constant, we set the learning rate of the gating network  $\eta_{moe}$  to 0.05, 0.1, and 0.5. As shown in Figure 7, for both  $S = 0$  and  $S = 20$  heterogeneous data settings, the MOE gating network should be assigned a higher learning rate. A smaller learning rate may cause the model to get stuck in local minima or saddle points, leading to worse performance.

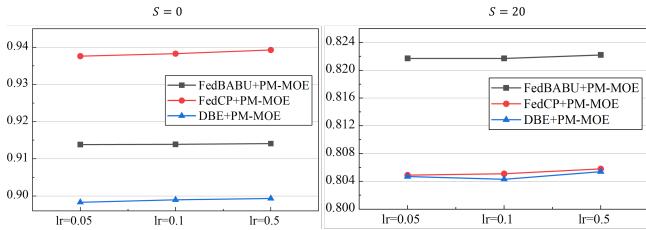


Figure 7: Impact of Gating Network Learning Rate in PM-MOE.

- Impact of MOE Training Iterations:** We further explored whether the number of local training iterations affects PM-MOE. As shown in Figure 8, after adding PM-MOE to three algorithms, performance slightly decreases with more training iterations, possibly due to overfitting. Therefore, in the case of converged pre-trained personalized federated learning, training for 50 epochs per client is sufficient.

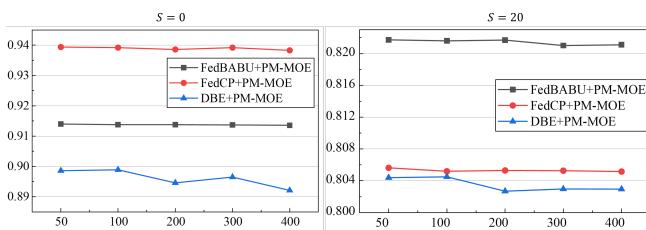


Figure 8: Impact of Training Epochs in PM-MOE.

#### 4.3 Analysis of the combination of personalized federated learning and MOE

For integrating MoE in personalized federated learning, both PFL-MoE and FedMoE use gated networks to adjust the weight balance between local personalized models and the global model. Both methods train the local models synchronously. For fair comparison, we employed a gated network to balance the global and local models'

Table 4: Moe Combination Analysis Results

Method	MNIST	FMNIST	Cifar10	Cifar100	TINY	AGNews
Fed-Syn-MoE	89.48	91.47	78.31	18.75	13.59	93.37
PM-MOE	99.85	98.61	93.95	70.68	66.33	94.76

weights, training it synchronously with gradient optimization, referred to as synchronous MoE. The pre-training phase lasted 2000 rounds, with results presented in Table 4. While performance degradation in synchronous MoE is minor for the 4-class AGNews [64] dataset, it becomes significant as task complexity increases, particularly for datasets with 10, 100, or 200 classes like MNIST [28], FMNIST [53], CIFAR-10 [25], CIFAR-100 [25], and TINY [7]. This decline likely occurs because synchronous training forces the gated network to balance unconvolved global and local parameters, making it more susceptible to noise and assigning suboptimal weights, which degrades overall model performance.

#### 5 Related Work

**Personalized Federated Learning and MOE.** In personalized federated learning, methods integrating Mixture of Experts [20, 65] (MoE) models, such as PFL-MoE [16] and FedMoE [57]. PFL-MoE primarily addresses homogeneous models, modulating the experts' weights via the gating network. In contrast, FedMoE emphasizes model heterogeneity by incorporating experts with more parameters than the global model to better capture local data characteristics.

**Energy-based denoising methods.** Energy-based models (EBMs) [29] assign scalar energy values to input configurations, capturing variable dependencies. They have been applied in generative modeling [12], out-of-distribution detection [14, 36], open-set classification [2], and incremental learning [51]. In personalized federated learning, EBMs quantify relationships between model parameters using energy as a metric. For Mixture of Experts (MoE) models, EBMs can filter ineffective experts, denoising the model. However, their use for expert denoising in MoE remains underexplored.

#### 6 Conclusion

In this article, we propose the PM-MOE framework to integrate the construction of a personalized parameter pool with local MOE training. PM-MOE aggregates the converged private model parameters from all clients, allowing each client to selectively reference the knowledge of others. This architecture effectively enhances the ability of model-splitting-based personalized federated learning algorithms to learn global knowledge. Through extensive experiments and theoretical analysis, we demonstrate the superiority of PM-MOE.

#### Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 62406036, the National Key Research and Development Program of China under Grant 2024YFC3308503, the Key Laboratory of Target Cognition and Application Technology under Grant 2023-CXPT-LC-005, and also sponsored by SMP-Zhipu.AI Large Model Cross-Disciplinary Fund under Grant ZPCG20241029322.

## References

- [1] Mohammed Adnan, Shivam Kalra, Jesse C Cresswell, Graham W Taylor, and Hamid R Tizhoosh. 2022. Federated learning and differential privacy for medical image analysis. *Scientific reports* 12, 1 (2022), 1953.
- [2] Mohamad M Al Rahhal, Yakkoub Bazi, Reham Al-Dayil, Bashair M Alwadei, Nassim Ammour, and Naif Alajlan. 2022. Energy-based learning for open-set classification in remote sensing imagery. *International Journal of Remote Sensing* 43, 15–16 (2022), 6027–6037.
- [3] Mohammad Al-Rubaie and J. Morris Chang. 2016. Reconstruction Attacks Against Mobile-Based Continuous Authentication Systems in the Cloud. *IEEE Trans. Inf. Forensics Secur.* 11, 12 (2016), 2648–2663. <https://doi.org/10.1109/TIFS.2016.2594132>
- [4] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated Learning with Personalization Layers. *CoRR* abs/1912.00818 (2019).
- [5] Nuria Rodríguez Barroso, Goran Stipčić, Daniel Jiménez-López, José Antonio Ruiz-Millán, Eugenio Martínez-Cámara, Gerardo González-Seco, María Victoria Luzón, Miguel Angel Veganzana, and Francisco Herrera. 2020. Federated Learning and Differential Privacy: Software tools analysis, the Sherpa.ai FL framework and methodological guidelines for preserving data privacy. *Inf. Fusion* 64 (2020), 270–292. <https://doi.org/10.1016/J.INFFUS.2020.07.009>
- [6] Hong-You Chen and Wei-Lun Chao. 2022. On Bridging Generic and Personalized Federated Learning for Image Classification. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- [7] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. 2017. A Downsampled Variant of ImageNet as an Alternative to the CIFAR datasets. *CoRR* abs/1707.08819 (2017). arXiv:1707.08819 <http://arxiv.org/abs/1707.08819>
- [8] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. Exploiting Shared Representations for Personalized Federated Learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 2089–2099.
- [9] Damaï Dai, Yutao Sun, Li Dong, Yaru Hao, Shumeng Ma, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559* (2022).
- [10] Lydia de la Torre. 2018. A guide to the california consumer privacy act of 2018. Available at SSRN 3275571 (2018).
- [11] Canh T. Dinh, Nguyen Hoang Tran, and Tuan Dung Nguyen. 2020. Personalized Federated Learning with Moreau Envelopes. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).
- [12] Yilun Du and Igor Mordatch. 2019. Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 3603–3613. <https://proceedings.neurips.cc/paper/2019/hash/378a063b8fdb1db941e34f4bde584c7d-Abstract.html>
- [13] Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar. 2020. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).
- [14] Jiawei Fan, Zhonghou Ou, Xie Yu, Junwei Yang, Shigeng Wang, Xiaoyang Kang, Hongxing Zhang, and Meina Song. 2022. Episodic projection network for out-of-distribution detection in few-shot learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 3076–3082.
- [15] Jiawei Fan, Yu Zhao, Xie Yu, Lihua Ma, Junqi Liu, Fangqiu Yi, and Boxun Li. 2022. DTR: An Information Bottleneck Based Regularization Framework for Video Action Recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*. 3877–3885.
- [16] Binbin Guo, Yuan Mei, Danyang Xiao, and Weigang Wu. 2021. PFL-MoE: Personalized Federated Learning Based on Mixture of Experts. In *Web and Big Data - 5th International Joint Conference, APWeb-WAIM 2021, Guangzhou, China, August 23-25, 2021, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12858)*, Leong Hou U, Marc Spaniol, Yasushi Sakurai, and Junyong Chen (Eds.). Springer, 480–486.
- [17] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated Learning for Mobile Keyboard Prediction. *CoRR* abs/1811.03604 (2018). arXiv:1811.03604 <http://arxiv.org/abs/1811.03604>
- [18] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29, 6 (2012), 82–97.
- [19] Muhammad Akbar Husnoo, Adnan Anwar, Nasser Hosseinzadeh, Shama Naz Islam, Abdun Naser Mahmood, and Robin Doss. 2022. Fedrep: Towards horizontal federated load forecasting for retail energy providers. In *2022 IEEE PES 14th Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. IEEE, 1–6.
- [20] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive Mixtures of Local Experts. *Neural Comput.* 3, 1 (1991), 79–87. <https://doi.org/10.1162/NECO.1991.3.1.79>
- [21] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, Mirella Lapata, Phil Blunsom, and Alexander Koller (Eds.). Association for Computational Linguistics, 427–431.
- [22] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kalista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D’Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Samni Koyejo, Tancrede Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* 14, 1-2 (2021), 1–210.
- [23] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 5132–5143.
- [24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4015–4026.
- [25] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
- [26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).
- [27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.
- [28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
- [29] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).
- [30] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-Contrastive Federated Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 10713–10722.
- [31] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and Robust Federated Learning Through Personalization. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 6357–6368.
- [32] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2021. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. arXiv:2102.07623 [cs.LG] <https://arxiv.org/abs/2102.07623>
- [33] Xin-Chun Li, De-Chuan Zhan, Yunfeng Shao, Bingshuai Li, and Shaoming Song. 2021. FedPHP: Federated Personalization with Inherited Private Models. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12975)*, Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and José Antonio Lozano (Eds.). Springer, 587–602.
- [34] Paul Pu Liang, Terrance Liu, Ziyan Liu, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Think Locally, Act Globally: Federated Learning with Local and Global Representations. *CoRR* abs/2001.01523 (2020). arXiv:2001.01523 <http://arxiv.org/abs/2001.01523>
- [35] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. 2020. Ensemble Distillation for Robust Model Fusion in Federated Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).

- [36] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2020. Energy-based Out-of-distribution Detection. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). <https://proceedings.neurips.cc/paper/2020/hash/f5496252609c43eb8a3d147ab9b9c006-Abstract.html>
- [37] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. 2020. FedVision: An Online Visual Object Detection Platform Powered by Federated Learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 13172–13179. <https://doi.org/10.1609/AAAI.V34I08.7021>
- [38] Bing Luo, Pengchao Han, Peng Sun, Xiaomin Ouyang, Jianwei Huang, and Ningning Ding. 2023. Optimization Design for Federated Learning in Heterogeneous 6G Networks. *IEEE Netw.* 37, 2 (2023), 38–43.
- [39] Jun Luo and Shandong Wu. 2022. Adapt or Adaptation: Learning Personalization for Cross-Silo Federated Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 2166–2173. <https://doi.org/10.24963/ijcai.2022/301>
- [40] Saeed Masoudnia and Reza Ebrahimpour. 2014. Mixture of experts: a literature survey. *Artificial Intelligence Review* 42 (2014), 275–293.
- [41] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Xiaojin (Jerry) Zhu (Eds.). PMLR, 1273–1282.
- [42] Jaehoon Oh, Sangmoock Kim, and Se-Young Yun. 2022. FedBABU: Toward Enhanced Representation for Federated Image Classification. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- [43] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.
- [44] Protection Regulation. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council. *Regulation (eu)* 679 (2016), 2016.
- [45] Hyowoon Seo, Jihong Park, Seungeun Oh, Mehdi Bennis, and Seong-Lyun Kim. [n. d.]. 16 federated knowledge distillation. ([n. d.]).
- [46] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Fei Wu, and Chao Wu. 2020. Federated Mutual Learning. *CoRR* abs/2006.16765 (2020). arXiv:2006.16765 <https://arxiv.org/abs/2006.16765>
- [47] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. 2022. FedProto: Federated Prototype Learning across Heterogeneous Clients. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 8432–8440.
- [48] Fengrui Tian, Jiawei Fan, Xie Yu, Shaoyi Du, Meina Song, and Yu Zhao. 2022. TCVM: Temporal Contrasting Video Montage Framework for Self-supervised Video Representation Learning. In *Proceedings of the Asian Conference on Computer Vision*. 1539–1555.
- [49] Michalis K Titsias and Aristidis Likas. 2002. Mixture of experts classification using a hierarchical mixture model. *Neural Computation* 14, 9 (2002), 2221–2244.
- [50] Praveen Venkateswaran, Vatche Isahagian, Vinod Muthusamy, and Nalini Venkatasubramanian. 2023. Fedgen: Generalizable federated learning for sequential data. In *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*. IEEE, 308–318.
- [51] Yabin Wang, Zhiheng Ma, Zhiwu Huang, Yaowei Wang, Zhou Su, and Xiaopeng Hong. 2023. Isolation and Impartial Aggregation: A Paradigm of Incremental Learning without Interference. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 10209–10217. <https://doi.org/10.1609/AAAI.V37I18.26216>
- [52] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. Communication-efficient federated learning via knowledge distillation. *Nature communications* 13, 1 (2022), 2032.
- [53] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR* abs/1708.07747 (2017). arXiv:1708.07747 <http://arxiv.org/abs/1708.07747>
- [54] Jian Xu, Xinyi Tong, and Shao-Lun Huang. 2023. Personalized Federated Learning with Feature Alignment and Classifier Collaboration. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- [55] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. 2019. *Federated Learning*. Morgan & Claypool Publishers. <https://doi.org/10.2200/S00960ED2V01Y201910AIM043>
- [56] Liping Yi, Gang Wang, Xiaoguang Liu, Zhuan Shi, and Han Yu. 2023. FedGH: Heterogeneous Federated Learning with Generalized Global Header. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, Abdulkotaleb El-Saddik, Tao Mei, Rita Cucchiara, Marco Bertini, Diana Patricia Tobon Vallejo, Pradeep K. Atrey, and M. Shamin Hossain (Eds.). ACM, 8686–8696.
- [57] Liping Yi, Han Yu, Chao Ren, Heng Zhang, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. 2024. FedMoE: Data-Level Personalization with Mixture of Experts for Model-Heterogeneous Personalized Federated Learning. *arXiv preprint arXiv:2402.01350* (2024).
- [58] Xie Yu and Wentao Zhang. 2024. Anchor-Based Masked Generative Distillation for Pixel-Level Prediction Tasks. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024*. BMVA. <https://papers.bmvc2024.org/0365.pdf>
- [59] Jianqing Zhang, Yang Hua, Jian Cao, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. 2023. Eliminating Domain Bias for Federated Learning in Representation Space. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.).
- [60] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, Jian Cao, and Haibing Guan. 2023. GPFL: Simultaneously Learning Global and Personalized Feature Information for Personalized Federated Learning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 5018–5028.
- [61] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. 2023. FedALA: Adaptive Local Aggregation for Personalized Federated Learning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, Brian Williams, Yiling Chen, and Jennifer Neville (Eds.). AAAI Press, 11237–11244.
- [62] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. 2023. FedCP: Separating Feature Information for Personalized Federated Learning via Conditional Policy. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (Eds.). ACM, 3249–3261.
- [63] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Jian Cao. 2023. PFLib: Personalized Federated Learning Algorithm Library. *CoRR* abs/2312.04992 (2023). <https://doi.org/10.48550/ARXIV.2312.04992> arXiv:2312.04992
- [64] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.), 649–657. <https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f38dc8b4be867a9a02-Abstract.html>
- [65] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Y. Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. 2022. Mixture-of-Experts with Expert Choice Routing. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sammi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).
- [66] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. 2021. Data-Free Knowledge Distillation for Heterogeneous Federated Learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 12878–12889.

## A Appendix

### A.1 Theoretical Derivations

PROOF. Considering  $M$  client experts predict independently, the probability that exactly  $s$  experts can predict correctly is given by the binomial distribution  $\text{Bin}(M, p)$ :

$$P(\mathcal{A}) = \binom{M}{s} p^s (1-p)^{M-s}, \quad (22)$$

where the event set  $\mathcal{A} := \{s \text{ out of } M \text{ experts can predict correctly}\}$ . Under the above condition, if the gate network can assign a sample to any of these  $s$  client experts, then the MPE can predict the sample correctly. Therefore, we have:

$$P(\mathcal{B} | \mathcal{A}) = \frac{(1+\alpha)s}{(1+\alpha)s + (M-s)} = \frac{(1+\alpha)s}{M+\alpha s}, \quad (23)$$

where  $\mathcal{B} := \{\text{MPE can predict correctly}\}$ . According to the law of total probability, the probability that the MPE can predict correctly is:

$$\begin{aligned} P(\mathcal{B}) &= \sum_{\mathcal{A}} P(\mathcal{A}) P(\mathcal{B} | \mathcal{A}) \\ &= \sum_{s=0}^M \binom{M}{s} p^s (1-p)^{M-s} \frac{(1+\alpha)s}{M+\alpha s} \\ &= \sum_{s=1}^M \binom{M-1}{s-1} p^{s-1} (1-p)^{M-s} \frac{(1+\alpha)Mp}{M+\alpha s} \\ &= \sum_{t=0}^{M-1} \binom{M-1}{t} p^t (1-p)^{M-1-t} \frac{(1+\alpha)Mp}{M+\alpha(t+1)} \quad (\text{Let } t = s-1) \\ &= \mathbb{E} \left[ \frac{(1+\alpha)Mp}{M+\alpha(t+1)} \right]. \end{aligned} \quad (24)$$

Here, the expectation is taken over  $t$  which follows  $\text{Bin}(M-1, p)$ . Define the function:

$$f(t) = \frac{(1+\alpha)Mp}{M+\alpha(t+1)}. \quad (25)$$

We observe that  $f(t)$  is a convex function with respect to  $t$  because its second derivative is positive:

$$f''(t) = \frac{2(1+\alpha)Mpa^2}{(M+\alpha t+\alpha)^3} > 0, \quad (26)$$

since  $\alpha, p, M > 0$ . By Jensen's inequality, we have:

$$\mathbb{E}[f(t)] \geq f(\mathbb{E}[t]). \quad (27)$$

The expected value of  $t$  is:

$$\mathbb{E}[t] = (M-1)p. \quad (28)$$

Combining (24), (27) and (28) yields

$$\begin{aligned} P_{\text{MPE}} &= P(\mathcal{B}) \geq f(\mathbb{E}[t]) = f((M-1)p) \\ &= \frac{(1+\alpha)Mp}{M+\alpha((M-1)p+1)} \quad (29) \\ &= \frac{(1+\alpha)p}{1+\alpha(p+\frac{1-p}{M})}. \end{aligned}$$

The last expression is strictly increasing with respect to  $\alpha$  when  $\alpha > 0$ , and thus:

$$P_{\text{MPE}} \geq \frac{(1+\alpha)p}{1+\alpha(p+\frac{1-p}{M})} > \frac{(1+0)p}{1+0(p+\frac{1-p}{M})} = p = P_{\text{client}}. \quad (30)$$

□

### A.2 Algorithm

Following the design principles of MPP, MPE and denoising module, we apply this algorithm to nine state-of-the-art model-splitting-based personalized federated learning algorithms. The general training process is outlined in Algorithm 1.

---

#### Algorithm 1 Personalized Model Training with MOE

---

**Input:**  $M$ : Number of clients;  $W_{g,fe}$ : Pre-trained parameters of the global feature extractor;  $W_{g,hd}$ : Pre-trained parameters of the global head;  $\{W_{p,fe}^j, W_{p,hd}^j\}_{j=1}^M$ : Pre-trained parameters of the client  $j$  personalized head;  $\eta_{moe}$ : Local MOE learning rate;  $k$ : Top- $k$  value for MOE weights;  $E_{moe}$ : Local MOE training iterations;  $\theta_G^j$ : Client  $j$ 's model parameters of the gating network;  $\mathcal{W}_{PE}$ ,  $\mathcal{W}_{PP}$ : Personalized expert set and personalized parameter set, respectively.  $\gamma$ : Dropout ratio.

**Output:**  $\{\theta_{PE}^1, \dots, \theta_{PE}^M\}$ ,  $\{\theta_{PP}^1, \dots, \theta_{PP}^M\}$ : Reasonable personalized gate network models.

1: Server collects all the client's personalized experts and parameters to form a personalized pool  $\mathcal{W}_{PE}$ ,  $\mathcal{W}_{PP}$  and sends them to  $M$  clients.  
2: Server sends  $W_{g,fe}$ ,  $W_{g,hd}$  to  $M$  clients.

3: **for**  $j \in [M]$  in parallel **do**

- 4: **local initialization**
- 5: Client  $j$  overwrites  $W_{g,fe}$ ,  $W_{g,hd}$  with the server parameters and freezes all of them.
- 6: Client  $j$  initializes the gated network  $\theta_{PE}^j$ ,  $\theta_{PP}^j$  for  $\mathcal{W}_{PE}$ ,  $\mathcal{W}_{PP}$  collected by the server.
- 7: The MOE combination of  $\{\theta_{PE}^j, \mathcal{W}_{PE}\}$ ,  $\{\theta_{PP}^j, \mathcal{W}_{PP}\}$  replaces the local personalized experts and parameters.

8: **local MOE learning**

9: **for**  $t = 0$  to  $E_{moe}$  **do**

- 10: Extract negative Helmholtz free energy  $H^k(h^j, h^k)$  by Eq.12
- 11: Removed  $\gamma$ -proportion part by Dropout( $\gamma$ ,  $\mathcal{W}_{PE}$ ,  $\mathcal{W}_{PP}$ )
- 12: Client  $j$  updates  $\theta_{PE}^j$ ,  $\theta_{PP}^j$  simultaneously:
- 13:  $\theta_{PE}^j \leftarrow \theta_{PE}^j - \eta_{moe} \nabla_{\theta_{PE}^j} G_{PE}^j$
- 14:  $\theta_{PP}^j \leftarrow \theta_{PP}^j - \eta_{moe} \nabla_{\theta_{PP}^j} G_{PP}^j$
- 15: **end for**
- 16: **end for**
- 17: **return**  $\{\theta_{PE}^1, \dots, \theta_{PE}^M\}$ ,  $\{\theta_{PP}^1, \dots, \theta_{PP}^M\}$

---

### A.3 Preliminary related work

*Personalized Federated Learning.* Personalized Federated Learning (PFL) was introduced to address the limitations of traditional federated learning in handling non-IID data and personalized requirements. PFL employs various strategies such as regularization [11, 31], meta-learning [13], knowledge distillation [45–47, 52, 54], model splitting [4], and personalized aggregation [33, 39, 61]. pFedMe [11] leverages the convexity and smoothness of Moreau Envelopes to facilitate its convergence analysis, while Per-FedAvg [13] incorporates meta-learning into federated learning. FedDistill [45] transfers global knowledge to local models through distillation.

*Model-Splitting-Based Personalized Federated Learning.* Model-splitting-based personalized federated learning has recently gained traction by balancing personalization and global consistency through model partitioning. These methods fall into three categories: the first combines personalized feature extractors with a globally shared classifier, as seen in FedGH[56] and LG-FedAvg [4], allowing clients to maintain unique feature extraction while ensuring consistency through a shared classifier. The second type uses a globally shared feature extractor and personalized classifiers, as demonstrated by FedBABU [42], FedCP [62], GPFL [60], FedPer [4], FedRep [8], and FedRod [6], enabling client-specific adaptation while preserving shared feature extraction. The third type, such as DBE [59], integrates local personalized parameters with shared feature extractors and classifiers, enhancing performance on individual clients.

#### A.4 Privacy Analysis

For model-splitting-based personalized federated learning algorithms combined with PM-MOE, data privacy is ensured in both phases.

**In the pre-training phase**, each client uploads only the shared parameters to the server, while personalized parameters are trained locally. Due to the model splitting, the link between shared and personalized parameters is severed. The gradient information of personalized parameters remains private to each client, making it difficult to breach data privacy through model inversion attacks [3].

**In the PM-MOE phase**, both the server and clients only receive the converged personalized model parameters. Clients cannot infer the training data or other private information from the model parameters. Therefore, the proposed approach effectively safeguards data privacy.

#### A.5 Experimental Details

To ensure fairness, we employ a 4-layer CNN model as the backbone for Cifar10, Cifar100, MNIST [28], Fashion-MNIST [53], and Tiny-ImageNet [7] datasets, and a fastText [21] model for AG News. Each personalized model is pre-trained for 2000 epochs until convergence. We optimize three key parameters:  $\eta_{moe}$  (local MOE learning rate),  $k$  (top k MOE weights), and  $E_{moe}$  (local MOE training iterations). All experiments are executed on a single RTX 3090 GPU.

#### A.6 Dataset and Data Partitioning.

We use public datasets to perform experiments and evaluate the performance of PM-MOE. Specifically, we adopt a Dirichlet distribution [35] with a shared ratio  $S(0 < S < 100)$  for data partitioning.

- **Dirichlet distribution with  $S = 20$ :** In the first setting, 20% of the data for each class is uniformly distributed among  $M$  clients, and the remaining data is assigned based on Dirichlet-distributed weights.
- **Dirichlet distribution with  $S = 0$ :** In the second setting, no constraints are placed on class distribution across clients, with all data allocated based on Dirichlet-distributed weights.

The detailed descriptions and statistics of these datasets are as follows:

- **MNIST [28]** dataset is a widely used collection for handwritten digit recognition, compiled by the National Institute of Standards

and Technology (NIST). It consists of 60,000 training images and 10,000 test images, each a 28x28 grayscale representation of digits from 0 to 9.

- **FMNIST [53]** is a dataset of fashion product images intended as a more challenging alternative to the traditional MNIST. It contains 10 categories of clothing items, such as T-shirts, trousers, and sweaters, with 7,000 grayscale images per category. There are 60,000 training images and 10,000 test images, all at 28x28 pixels. Fashion MNIST presents a greater challenge in terms of image quality and diversity, featuring more background details and varying perspectives.
- **Cifar10 [25]** consists of 60,000 32x32 color images divided into 10 classes, with 6,000 images per class. Of these, 50,000 are used for training and 10,000 for testing. The dataset is split into five training batches and one test batch, each containing 10,000 images. The test batch includes 1,000 randomly chosen images from each class, while the training batches may have varying class distributions across batches.
- **Cifar100 [25]** dataset contains 60,000 32x32 color images, but it is divided into 100 classes, with 600 images per class. Each class has 500 images for training and 100 for testing. These 100 classes are grouped into 20 super-classes, with each image having both a "fine" label (its specific class) and a "coarse" label (its super-class).
- **TINY [7]** dataset is a subset of ImageNet, released by Stanford University. It comprises 200 classes, each with 500 training images, 50 validation images, and 50 test images. The images have been preprocessed and resized to 64x64 pixels and are commonly used in deep learning for image classification tasks.
- **AGNews [64]** dataset is an open dataset for text classification, containing 120,000 news headlines and descriptions from four categories: World, Sports, Business, and Technology. Each category includes 30,000 samples, with 120,000 samples in the training set and 7,600 in the test set.

#### A.7 Baselines

In our experiments, the comparison baselines mainly include traditional federated learning methods (FedAvg, FedProx, SCAFFOLD, MOON, and FedGen), federated learning of personalized experts (FedGH, LG-FedAvg, FedBABU, FedCP, GPFL, FedPer, FedRep, FedRod), and federated learning of personalized parameters (DBE).

- **FedAvg [41]** is a pioneering algorithm in federated learning. Its core idea is to send the global model from the server to participating clients, where each client trains the model using their local data. The updated model parameters are then uploaded to the server, which computes the average of these parameters to update the global model. FedAvg can encounter performance bottlenecks when faced with highly imbalanced data or significant differences in client computing power.
- **FedProx [38]** aims to address the performance degradation of FedAvg when dealing with non-i.i.d. data. It introduces a regularization term during local training to penalize the deviation

- of model parameters from the global model, stabilizing the optimization process and preventing local models from straying too far from the global model.
- **SCAFFOLD** [23] tackles the issue of client drift by using control variates to reduce the variance between local updates and the global model. This ensures closer alignment between local models and the global objective, especially in non-i.i.d. data scenarios.
  - **MOON** [30] is a federated learning algorithm based on contrastive learning. It aims to minimize the feature representation difference between the local and global models while maximizing the difference between successive local models. By contrasting the global and local model representations, MOON enhances the generalization ability of the global model in federated environments.
  - **FedGen** [66] is a federated learning algorithm using knowledge distillation without data. It employs a lightweight generator on the server side to synthesize data, which is broadcasted to clients to assist their model training. This method not only optimizes the global model but also introduces inductive bias to local models, improving generalization in non-i.i.d. settings.
  - **FedGH** [56] is a federated learning framework for heterogeneous models. It trains a shared Global Prediction Header (GPH) to integrate diverse model structures from different clients. The GPH is trained using feature representations extracted by clients' private feature extractors and learns global knowledge from various clients. The server then transmits the shared GPH to all clients, replacing their local prediction heads.
  - **LG-FedAvg** [34] is a variant of FedAvg that trains both global and local models simultaneously. The global model acts as a classifier, while the local model is a feature extractor. During each iteration, both the classifier and feature extractor are updated concurrently without freezing any part of the model.
  - **FedBABU** [42] updates only the body of the model during training, leaving the head randomly initialized and unchanged. This allows the global model to improve generalization during training, while the head is fine-tuned for personalization during evaluation, achieving efficient personalization with consistent performance improvements.
  - **FedCP** [62] introduces conditional layers tailored to each client's data, which split the output of a shared extractor into personalized and global representations. The shared classifier handles global representations, while personalized classifiers manage personalized ones. Additionally, FedCP sets a regularization loss, ensuring that global feature representations remain as consistent as possible across rounds.
  - **GPFL** [60] personalizes federated learning by incorporating personalized layers into the global model, capturing client-specific features. GPFL aims to adapt to each client's unique needs while maintaining privacy, making it suitable for scenarios with highly heterogeneous data distributions.
  - **FedPer** [4] personalizes federated learning by keeping certain model layers (typically the final few) private to each client while sharing the remaining layers globally. This enables each client to

fine-tune their local layers for personalized tasks while benefiting from the shared global model. Unlike FedBABU, the local classification head in FedPer is not frozen, and both the feature extractor and classification head are optimized during local training.

- **FedRep** [8] first learns a shared representation through a matrix method, followed by alternating updates between clients and the server. FedRep demonstrates strong convergence in multilinear regression problems and significantly reduces sample complexity for new clients joining the system.
- **FedRod** [6] introduces a robust loss function that allows clients to train a universal predictor on non-identically distributed categories. It also includes a lightweight adaptive module (personalized classifier) that minimizes each client's empirical risk based on the shared universal predictor.
- **DBE** [59] is a method designed to tackle data heterogeneity in federated learning. It eliminates domain shifts in the representation space, optimizing the bidirectional knowledge transfer process between the server and clients. DBE sets a group of locally optimized private parameters to align and correct global model discrepancies.

## A.8 Evaluation Metrics

In personalized federated learning, the assessment of global accuracy can be formulated as the weighted sum of each client's accuracy rate multiplied by its sample proportion. The formal expression is as follows:

$$A_{total} = \sum_{j=1}^M \frac{N^j}{N} \cdot A^j \quad (31)$$

where  $A_{total}$  denotes the weighted total accuracy.  $M$  is the total number of clients.  $A^j$  represents the accuracy of the  $j$ -th client, and  $N^j$  is the number of samples from the  $j$ -th client.  $N = \sum_{j=1}^M N^j$  is the total number of samples across all clients.  $\frac{N^j}{N}$  signifies the proportion of samples from the  $j$ -th client.

## A.9 Concerns about time cost

The performance here refers to the average improvement across all datasets. For instance, on the MNIST dataset, it has already exceeded 99.80%. We calculate the improvement by summing the values across all datasets and dividing by the total number of datasets, which may make the performance gains appear less significant.

In our ablation study (Table 2), the proposed PM-MOE architecture applied to the recent state-of-the-art DBE algorithm shows an improvement of 3.28%, particularly on the FMNIST algorithm.

We also have case studies to address your concerns about the local MOE's time consumption. Since local fine-tuning adjusts only a small number of gating network parameters, PM-MOE typically requires just 50 iterations, resulting in minimal overall time consumption.

**Case:** The major computational burden lies in the pre-training phase. For instance, using FedCP on AGNews:

- FedCP (Pre-training): 53 hours.
- PM-MOE fine-tuning: extra 12.68 minutes (**0.39%** of pre-training).