This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

FedMPS: Federated Learning in a Synergy of Multi-Level Prototype-Based Contrastive Learning and Soft Label Generation

Wenxin Yang, Xingchen Hu, Xiubin Zhu, Rouwan Wu, Witold Pedrycz, *Life Fellow, IEEE*, Xinwang Liu, *Senior Member, IEEE*, and Jincai Huang

***Abstract*—Federated learning (FL) facilitates collaborative training among multiple clients while preserving data privacy by eliminating raw data transmission. However, the inherent data heterogeneity among participants induces bias during collaborative learning, significantly degrading the performance of local models. Existing FL solutions face critical challenges in achieving efficient knowledge transmission, particularly with respect to insufficient information extraction or excessive communication costs, which result in slow convergence and inferior performance. To address these limitations, we propose a novel FL framework in a synergy of multi-level prototype-based contrastive learning (CL) and soft label generation, named FedMPS. The proposed method first constructs multi-level prototypes from different layers of the model to capture semantic information in high-level features and detailed information in low-level features. These prototypes are then utilized through CL to enhance intra-class discriminability and intra-class consistency in the feature space. In addition, a prototype-guided soft label generation module is introduced to model latent interclass relationships in the output space. Instead of exchanging model parameters, FedMPS transmits only prototypes and soft labels, effectively reducing global knowledge shift and communication costs. Extensive experimental studies on six publicly available datasets validate the effectiveness of the proposed method when compared to the current state-of-the-art FL approaches. The code is available at github.com/wenxinyang1026/FedMPS**

***Index Terms*—Contrastive learning (CL), data heterogeneity, federated learning (FL), multilevel prototypes, soft label learning.**

Received 23 May 2024; revised 6 April 2025 and 12 July 2025; accepted 9 September 2025. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62376279. *(Corresponding author: Xingchen Hu.)*

Wenxin Yang and Jincai Huang are with the Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410073, China.

Xingchen Hu is with the Laboratory for Big Data and Decision and the School of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: xhu4@ualberta.ca).

Xiubin Zhu is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: xbzhu@mail.xidian.edu.cn).

Rouwan Wu is with the School of Systems Engineering, National University of Defense Technology, Changsha 410073, China.

Witold Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada, also with the Systems Research Institute, Polish Academy of Sciences, 00-901 Warsaw, Poland, and also with the Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Istinye University, Sariyer, 34396 Istanbul, Türkiye (e-mail: wpedrycz@ualberta.ca).

Xinwang Liu is with the School of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: xinwangliu@nudt.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2025.3611832

NOMENCLATURE

Notation	Description
N	Number of clients.
D_i	Local dataset of client C_i.
level	Level of features or prototypes.
z_i^{level}	Corresponding level features of client C_i.
p_i^{level}	Corresponding level prototypes of client C_i.
$\bar{p}^{\text{level}}$	Corresponding level global prototypes.
q_i	Local soft labels of client C_i.
$\bar{q}$	Global soft labels.
$f_i^{\text{level}}(\phi_i^{\text{level}};\cdot)$	Corresponding level encoder of client C_i.
$f_i(\psi_i;\cdot)$	Output layer of client C_i.
$f(\bar{\psi};.)$	Global model.

I. INTRODUCTION

FEDERATED learning (FL) aims to enable collaborative training without directly transferring raw data across multiple devices or institutions [1]. It achieves this by aggregating knowledge from distributed clients, improving model performance while maintaining data locality. The iterative phase of FL involves two fundamental stages: 1) each individual client updates its local model training on its private dataset and 2) the central server aggregates knowledge from various clients to facilitate the local model optimization. Recent studies have demonstrated the application of FL across various domains, including intelligent healthcare [2], [3], the Internet of Things (IoT) [4], [5], and object recognition [6], [7], [8].

An important challenge faced by FL is enabling clients to learn effective representations under data heterogeneity. Data heterogeneity refers to the different data distributions among individual clients, namely non-i.i.d. data. One solution is to *add constraints on the local model* based on FedAvg [9] to enhance its robustness [10], [11], [12], [13]. They realize the interaction of knowledge by exchanging model parameters. However, this parameter-based aggregation introduces challenges such as model bias and substantial communication overhead. Therefore, other studies [14], [15], [16], [17] have focused on *prototype-based aggregation*, which reduces communication costs and alleviates the problem of data heterogeneity. *Multi-prototype FL* has gained attention. Some studies utilize clustering to produce multiple prototypes

2162-237X © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

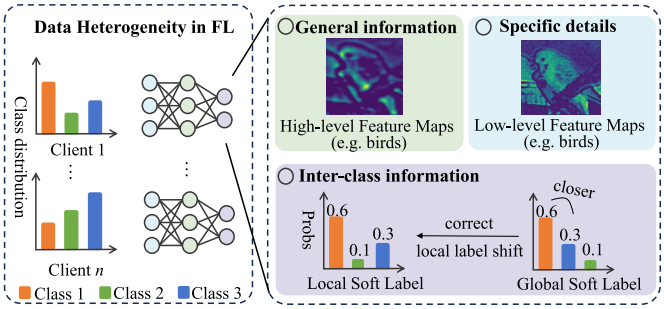


Fig. 1. Illustration of data heterogeneity in FL and the motivation for FedMPS. The left part shows that clients have different class distributions. The right part depicts the hierarchical knowledge framework employed by FedMPS: 1) low-level feature maps (visualized using LayerCAM [22]) with detailed information; 2) high-level feature maps (visualized using LayerCAM [22]) with general information; and 3) soft labels with global inter-class information.

for each class [18], [19], while others construct a semantic prototype for multiple classes [20]. However, these methods not only rely exclusively on deep features, but also bring about blurred classification boundaries between prototypes of similar classes. Moreover, the attempt to align multi-level prototypes through auxiliary branches [21] fails to ensure inter-class discriminability, while inevitably increasing communication and local storage overhead. These limitations highlight a critical challenge: existing methods either fail to extract sufficient and discriminative knowledge or impose excessive burdens, both of which constrain their performance and efficiency in non-i.i.d. scenarios.

In this study, as shown in Fig. 1, we tackle the challenge of data heterogeneity in FL from a unique perspective, inspired by an intuitive observation: integrating general information with specific details enhances comprehension more effectively than a unilateral focus, and latent inter-class information provides global insights to mitigate label shift in local clients. Therefore, we propose an FL framework, named *FedMPS*, which integrates multi-level prototype-based contrastive learning (CL) with a soft label generation module. On the local client side, consistency between multi-level global prototypes and corresponding local features is maximized to update the local model. This facilitates the full extraction and combination of low-level detailed information (specific details) and high-level semantic information (general information) to assist classification tasks. On the server side, local prototypes received from various clients are aggregated to update global prototypes. Simultaneously, these local prototypes are used to generate global soft labels, which are then distributed to clients for alignment with local soft labels. This enables local models to acquire more knowledge across classes and enhance their generalization capabilities. Overall, FedMPS offers a straightforward and effective approach to improving performance under data heterogeneity while controlling communication costs by enhancing information diversity based on prototypes.

The main contributions are summarized as follows.

1) A multi-level feature-prototype CL module is designed to capture both low-level detailed information and high-level semantic information, which further enhances inter-class discriminability and intra-class consistency, while considering communication overhead.
2) A soft label generation module is implemented on the server side, utilizing local prototypes to produce global soft labels, which helps mitigate the incomplete inter-class information caused by label shift in local models.
3) A large number of experiments based on six public datasets are conducted to evaluate the effectiveness and efficiency of FedMPS in heterogeneous data classification.

The remainder of this article is organized as follows. Section II provides a review of relevant literature. Section III details the proposed FedMPS algorithm. Experimental results are analyzed in Section IV and conclusions are drawn in Section V.

II. Related Work

In this section, we present a concise review of related work regarding this study, covering prototype learning, CL, and FL.

A. Prototype Learning

Prototype learning makes the features of samples more discriminative. It was first proposed by Reed [23], who defined the mean of a class in latent space as its prototype, and then assigned the sample to the nearest prototype class. Subsequently, prototype learning is applied to few-shot learning. For example, prototypical networks [24] combine the classic mean-of-class prototype learner with few-shot learning, which reflects a simpler inductive bias under limited data. This approach has been extended to more complex tasks, such as few-shot semantic segmentation [25], [26], few-shot class-incremental learning [27], and few-shot relation classification [28]. They all enhance representation learning through the semantics of prototypes in limited data scenarios. In practical FL scenarios, each client often has access to a limited amount of data [16], [29], which aligns with the few-shot learning discussed here. Therefore, this study adopts the classic mean-of-class prototype learning in FL due to its simplicity and efficiency.

Prototype learning has also been applied to representation learning in heterogeneous domains, further extending the data heterogeneity challenge addressed in this study. BHPL [30] introduced an end-to-end bidirectional heterogeneous prototype learning framework. Zhou et al. [31] tackled the multi-source domain adaptation problem through domain-invariant prototypes and domain-specific prototypes. These methods demonstrate the adaptability of prototype learning in capturing discriminative features across diverse data. Therefore, prototype learning can serve as a powerful tool for FL in heterogeneous data scenarios by extracting distinctive information for local model training.

B. Contrastive Learning

CL, an increasingly prominent technique in representation learning, effectively enhances sample discrimination in feature space. It starts with unsupervised learning. Hadsell et al. [32]

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

first proposed the concept of CL loss, a separate loss function for similar and dissimilar pairs. A representative CL framework called SimCLR [33] pulls augmented views of the same sample closer while pushing apart different samples. However, this instance-wise discrimination approach may inadvertently push apart negative samples with similar semantics. To address this issue, PCL [34] employs K-means clustering to encourage representations to be closer to their assigned prototypes.

Subsequently, SupCon [35] extends CL to a fully supervised setting. In this approach, samples from the same class are considered as positive pairs, while those from different classes are regarded as negative pairs. PaCo [36] introduces a set of parametric class-wise learnable centers to better handle imbalanced scenarios in supervised CL. SelfCon [37] utilizes a subnetwork to generate multiple outputs, ensuring consistent label information in a batch. Consequently, supervised CL has become a widely adopted approach for enhancing the discriminative power of feature representations. Interestingly, FL also brings new attempts to supervised CL. The aggregated knowledge from different clients forms a new type of augmented sample in traditional CL. The integration of collaborative information in FL and discriminative improvement in CL will mutually reinforce both techniques, driving further advancements in these fields.

C. Federated Learning

FL is a specific distributed learning paradigm that enables collaborative learning across clients while protecting data privacy. FedAvg [9] serves as the foundational work in FL, which aggregates local model parameters through a weighted average to obtain the global model parameters. However, data heterogeneity poses a significant challenge for traditional FL due to varying data distributions across clients. Subsequently, numerous efforts have been made to address it. One major approach is to impose constraints on local model updates. Building upon FedAvg [9], FedProx [10], SCAFFOLD [38], FedDyn [39], and FedDC [40] introduce auxiliary terms to correct model drift. While these methods are simple to implement, parameter-based transmission incurs high communication costs and slow convergence. Additionally, several advancements, including federated prototype learning, federated CL, and knowledge distillation-based FL, have emerged to improve performance when faced with the data heterogeneity challenge.

Federated prototype learning leverages prototypes as abstract knowledge in FL. Some approaches utilize a single-level prototype as unbiased knowledge to regularize local models. CCVR [15] constructs virtual representations based on local prototypes to refine the classifier. FedProto [16] transmits prototypes instead of gradients, significantly reducing communication overhead. However, these methods rely on a single form of prototypes, which may limit performance improvement. Therefore, multiple prototypes have been integrated into FL. MPFed [19] employs clustering to generate multiple prototypes per class, which can cause prototypes of similar classes to become closer. FedMLP [20] groups prototypes using K-means to obtain semantic-level prototypes that encompass multiple classes. While these prototypes based on deep features help minority classes with fewer samples to learn similar feature representations, they also increase the difficulty in distinguishing similar classes. Inspired by FedMLB [44], FedMBP [21] utilizes multi-level branches to align local prototypes with global ones, failing to consider inter-class discriminability and increasing communication and local storage overhead.

TABLE I
COMPARISON OF RELATED FL APPROACHES

Method	Multi-level information	Inter-class information	Efficient communication
FedProto [16]	✗	✗	✓
MPFed [18]	✓	✗	✗
FedMLP [20]	✓	✗	✗
FedMBP [21]	✓	✗	✗
MOON [11]	✗	✗	✗
FedProc [12]	✗	✗	✗
FedAUX [41]	✗	✓	✗
FedGEN [42]	✗	✗	✗
FedNTD [43]	✗	✓	✗
FedMPS (ours)	✓	✓	✓

Federated CL aims to obtain more consistent information sharing by combining CL with FL. MOON [11], a model-based federated CL approach, constructs positive sample pairs by associating local features with those from the global model, while using local features from the previous round as negative sample pairs. FedCOMO [45] extends the negative sample pairs through momentum-based parameter control, building on MOON [11]. Some studies further combine the abstract knowledge of prototypes with the enhanced discriminative power of CL. FedProc [12] introduces a global prototype contrastive loss function to regulate individual model updates. MPFedCL [18] leverages multiple prototypes obtained through K-means to represent each class, which then engage in CL with client features to refine local model updates. These methods achieve improvements with the consistent information of prototypes, but they still face inherent limitations, including slow convergence and degraded performance with insufficient prototype information.

Knowledge distillation-based FL transfers the knowledge of teacher models to student models in a compact form. While some approaches employ auxiliary datasets [41], [46], [47] to enhance generalization under data heterogeneity, their performance deteriorates when there is a distribution mismatch between the auxiliary and local datasets. To mitigate this issue, data-free methods have been explored through two primary strategies. The first involves generator-based approaches that synthesize artificial features [42] or samples [48] for distillation, though these methods incur high computational costs and are highly dependent on generation quality. The second strategy employs the aggregated model as a teacher [43], [49], [50], [51], introducing challenges of model drift and communication inefficiency due to frequent parameter exchanges.

We compare the aforementioned three types of methods with our work in Table I. Our work not only integrates multi-level

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

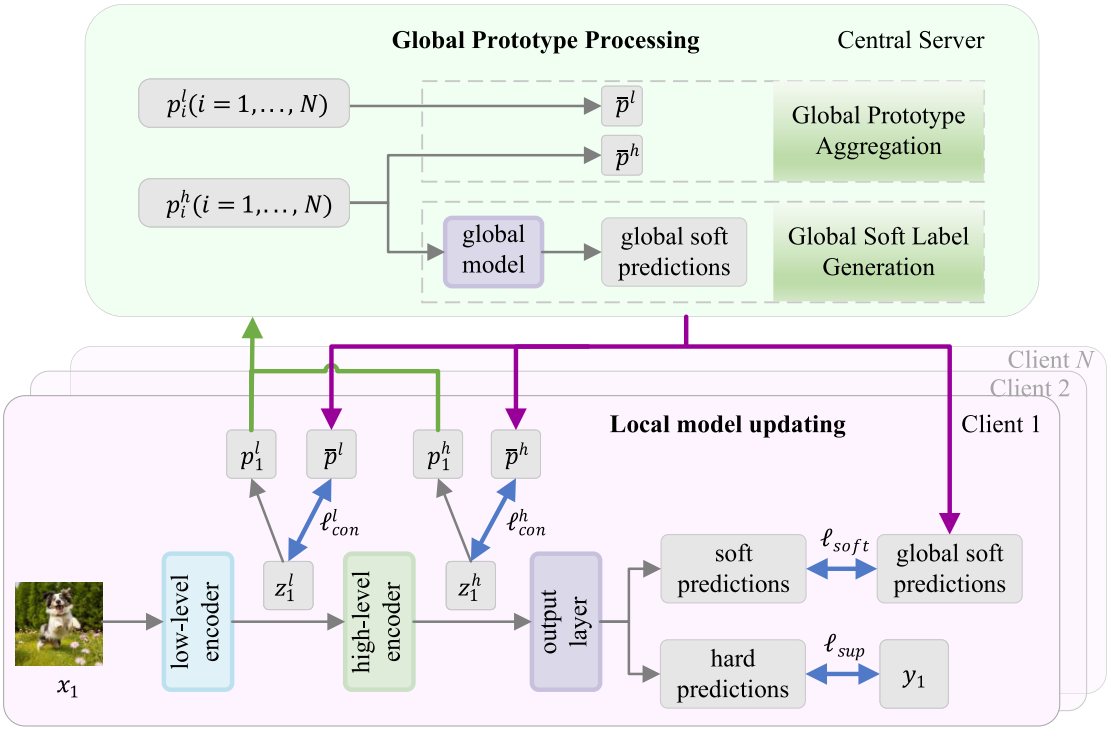


Fig. 2. Overview of FedMPS. First, each client updates its local model by minimizing the classification loss ℓ_{sup}, the multilevel CL loss consisting of ℓ_{con}^h and ℓ_{con}^l between the local features and global prototypes, and the KL divergence loss ℓ_{soft} between local soft predictions and global soft predictions. Then the clients send their multi-level local prototypes (→) averaged by feature vectors to the central server. The server generates global prototypes and global soft labels(→) based on local prototypes and returns them to local clients to assist in updating local models.

and inter-class relationship mining of prototype knowledge, but also considers the communication efficiency of alternative parameter transmission.

III. Proposed Algorithm: FedMPS

In this study, we propose an FL framework (FedMPS) in a synergy of multi-level prototype-based CL and soft label generation to address the performance limitation of FL under data heterogeneity. Next, we present the problem statement and outline the overall framework, followed by detailed discussions on local model updating and global prototype processing. The Nomenclature describes the main notations used in this study.

A. Problem Statement

Suppose there are N clients, denoted as $\{C_1, C_2,...,C_N\}$. Each client C_i has a local private dataset D_i sampled from the distribution $\mathbb{P}_i$. The distribution $\mathbb{P}_i$ varies across clients, reflecting data heterogeneity in FL. In general, each client builds a local model, which has the same structure as models from other clients. Our goal is to optimize the parameters w_i of local models constructed on each client C_i by conducting collaborative training through a central server, which enables learning without sharing the original data. For each client C_i, the training objective is to minimize its empirical loss $L_i(w_i) = \mathbb{E}_{(x,y)\sim D_i}[\ell_i(w_i;(x,y))]$, where $\ell_i(w_i;(x,y))$ is the loss function. Then, the overall objective function is defined as follows:

$$\arg\min_{w_1,w_2,\ldots,w_N} \sum_{i=1}^{N} \frac{|D_i|}{|D|} L_i(w_i) \tag{1}$$

where D represents the union of all local datasets, namely $D \triangleq \bigcup_{i=1}^{N} D_i$.

B. Overall Framework

An overview of the proposed framework is shown in Fig. 2, comprising two stages: local model updating and global prototype processing. In the first stage of local model updating, each client extracts low-level features, high-level features, soft predictions, and hard predictions after processing data through the model. The model is then updated by incorporating three loss components: multi-level CL loss between local features and global prototypes, soft label loss between local and global soft labels, and classification loss between hard predictions and true labels. Next, local multi-level prototypes, computed by averaging features, are sent to the central server. Subsequently, the process moves to the next stage of global prototype processing. The server updates global prototypes by aggregating local prototypes received from clients. Additionally, it generates global soft labels using a network structurally identical to the local model's output layer. Finally, the global prototypes and soft labels are distributed to each client for the next round of training until the convergence criteria are met.

By innovatively utilizing multi-level feature-prototype CL during the update of the local model, the extraction of both fine-grained low-level detailed information and high-level semantic information can be significantly enhanced. This facilitates collaborative learning to enforce class consistency across different clients. The central server aggregates the knowledge of local prototypes and effectively integrates global soft label knowledge derived from these prototypes, thereby capturing

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

latent inter-class relationships at a global scale and alleviating label shift in local models. Consequently, the diversity of collaborative information is enhanced, and the generalization capability of FL with heterogeneous data is improved.

C. Local Model Updating

In this study, to address data heterogeneity among clients while ensuring dimensional consistency for server aggregation, each client's local model follows the same network structure [9], [11], [12], composed of three parts: a low-level encoder $f_i^l(\phi_i^l;\cdot)$, a high-level encoder $f_i^h(\phi_i^h;\cdot)$, and an output layer $f_i(\psi_i;\cdot)$, where ϕ_i^l, ϕ_i^h, and ψ_i denote the weights of the low-level encoder, high-level encoder, and output layer of client C_i, respectively. During the forward pass, an input x is first processed by the low-level encoder to produce the feature embedding $z_i^l = f_i^l(\phi_i^l; x)$. This embedding is then fed into the high-level encoder to generate the high-level semantic feature $z_i^h = f_i^h(\phi_i^h; z_i^l)$. Finally, the output layer produces both hard and soft classification predictions.

Therefore, the local update loss of client C_i comprises three components

$$\begin{aligned}\ell_i\left(w_i;(x,y)\right) = {} & \ell_{\text{sup}}\left(w_i;(x,y)\right) + \lambda\ell_{\text{mcon}}\left(w_i;(x,y,\bar{p})\right) \\ & + \mu\ell_{\text{soft}}\left(w_i;w;\left(x,y,\bar{p}^h\right)\right)\end{aligned} \tag{2}$$

where w_i denotes the local model weight of client C_i and w represents the global model weight. $\bar{p}$ represents global prototypes, which include high-level global prototypes $\bar{p}^h$ and low-level global prototypes $\bar{p}^l$. The first term, ℓ_{sup}, is the standard cross-entropy loss used for classification tasks. The second term, ℓ_{mcon}, represents the multi-level CL loss between local features and global prototypes. The third term, ℓ_{soft}, measures the similarity between local soft labels and global soft labels. λ and μ are hyperparameters that control the relative importance of ℓ_{mcon} and ℓ_{soft}, respectively.

The multi-level CL loss ℓ_{mcon} is employed to facilitate the extraction of high-level semantic features and low-level detailed features. Both levels of features are subjected to supervised CL with their respective global prototypes. For a local training batch containing B samples on a single client, the local feature vectors z_i^{level} and their global prototype vectors $\bar{p}^{\text{level}}$ corresponding to their respective classes are collected into a set V^{level}, that is, $V^{\text{level}} \equiv \{z_i^{\text{level}}\} \bigcup \{\bar{p}^{\text{level}}\}$, where level $\in \{l, h\}$ represents the low- and high-level vectors, respectively. For each element in V^{level}, it is desirable to be closer to vectors of the same class, including both global prototypes and local features, while being farther from the vectors of different classes. This promotes the alignment of class-consistent information across different clients. Based on Supcon [35], the formula is expressed as follows:

$$\ell_{\text{con}}^{\text{level}} = \frac{1}{|I|}\sum_{i\in I}\frac{-1}{\left|R_{(i)}\right|}\sum_{r\in R_{(i)}}\log\frac{\exp\left(v_i^{\text{level}}\cdot v_r^{\text{level}}/\tau_1\right)}{\sum_{a\in A_{(i)}}\exp\left(v_i^{\text{level}}\cdot v_a^{\text{level}}/\tau_1\right)} \tag{3}$$

where level $\in \{l, h\}$ represents low- and high-level CL losses, respectively. Let $i \in I \equiv \{1,\ldots,2B\}$ be the index of each element in the collection V^{level}, where V^{level} contains B samples in a batch and their corresponding B global prototypes with the same labels, giving $|V^{\text{level}}| = 2B$. $A_{(i)} \equiv I \setminus \{i\}$ denotes all vector indices in the set V except i, and $R_{(i)} \equiv \{r \in A_{(i)} : y_r = y_i\}$ represents the positive vectors indices of the ith vector. The symbol $\cdot$ denotes the inner (dot) product and τ_1 is a temperature parameter. Unlike FedProc [12], our approach incorporates positive samples from both local feature vectors z_i^{level} and global prototypes $\bar{p}^{\text{level}}$, rather than relying solely on the global prototypes. This allows for leveraging information between client samples. Moreover, instead of constructing global prototypes from all client classes [12], our approach employs global prototypes derived only from the classes present in the current client. This approach helps mitigate overfitting and computational overhead.

Therefore, multi-level CL loss is formulated as a weighted sum of the low- and high-level contrastive losses, which are computed based on the local features and the global prototypes

$$\ell_{\text{mcon}} = \alpha\ell_{\text{con}}^l + \beta\ell_{\text{con}}^h \tag{4}$$

where α and β are hyperparameters to control the weights of low- and high-level contrastive loss, respectively.

When the high-level features z_i^h reach the final output decision layer $f_i(\psi_i;\cdot)$, they are subjected to classification through the traditional softmax function to produce hard predictions. Simultaneously, by adjusting the temperature coefficient τ_2, the local model produces soft labels. The soft label q_i is calculated in the following way:

$$q_i = \text{softmax}\left(\frac{f_i\left(\psi_i; z_i^h\right)}{\tau_2}\right). \tag{5}$$

When the temperature coefficient τ_2 is equal to 1, it is a general softmax function, yielding hard labels. But as the temperature coefficient τ_2 grows moderately, the probabilities of previously suppressed classes, which were initially close to zero, become nonnegligible. By leveraging the relative probability distribution, soft labels can capture latent knowledge among classes.

The global soft labels $\bar{q}$ from the server contain local prototype knowledge of different clients. Aligning the global and local soft labels encourages the local model to learn more inter-class information in an aggregated way, thereby mitigating the limited class knowledge caused by label shift. Therefore, the third component of the loss in local training is the Kullback–Leibler (KL) divergence loss of global and local soft labels. The formula can be expressed in the following manner:

$$\ell_{\text{soft}} = D_{\text{KL}}\left(\bar{q}\,\|q_i\right) = \sum_c \bar{q}_c\log\frac{\bar{q}_c}{q_{i,c}} \tag{6}$$

where c represents the index of each class.

Finally, the local client obtains the low-level local prototypes p_i^l and high-level local prototypes p_i^h by calculating the average values of the low-level features z_i^l and high-level features z_i^h of each class and then transmits them to the central server.

D. Global Prototype Processing

After receiving the class prototypes from all local clients, the server performs two tasks: global prototype aggregation and global soft label generation. The first task calculates the mean of prototypes for every class to form global prototypes.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Compared to gradient-based communication, this approach exhibits higher communication efficiency and is more suitable for heterogeneous scenarios. The second task takes the local high-level prototypes from different clients as input, generating global soft labels through a global model that shares the same structure as the output layer of each local model. By combining these two components, the complementary information from different clients is leveraged to enhance the generalization performance of each local model.

Following FedProto [16], the global prototype aggregation strategy computes global prototypes by aggregating local prototypes. For class k at each feature level, the global prototype $\bar{p}_k^{\text{level}}$ is generated by averaging local prototypes of class k from clients possessing this class. This hierarchical aggregation ensures that both class-specific and level-wise features are preserved for local model updating, while mitigating the bias of knowledge aggregation.

The global soft label generation module captures inter-class relationships based on prototypes to mitigate local label shifts. Since low-level prototypes increase model complexity and impair both computational efficiency and convergence, we employ high-level local prototypes for soft label generation. Let $p^h \triangleq \{p_i^h\}_{i\in[N]}$ represents the collection of high-level local prototypes from all N clients, which serves as input to the global model $f(\bar{\psi};\cdot)$. This model replicates the output layer structure $f_i(\psi_i;\cdot)$ of the local model to maintain consistency while preserving simplicity. The global model is optimized using standard cross-entropy loss as follows:

$$\ell_{\text{global}} = \ell_{\text{sup}}\left(f\left(\bar{\psi}; p^h\right), \bar{y}\right) \tag{7}$$

where $\bar{y}$ denotes the true labels corresponding to the local prototypes p^h. After updating the global model, the global soft labels for each class, enriched with inter-class information, are generated as follows:

$$\bar{q}_k = \frac{1}{|S_k|}\sum_{i\in S_k}\text{softmax}\left(\frac{f\left(\bar{\psi}; p_{i,k}^h\right)}{\tau_2}\right) \tag{8}$$

where $\bar{q}_k$ represents the global soft predictions of class k. S_k contains the indices of all clients possessing class k.

Finally, the updated global multi-level prototypes and soft predictions, which aggregate knowledge collaboratively learned from various clients, are distributed to clients for interaction with local models in the next round.

Algorithm 1 presents the overall process of FedMPS. In each communication round, the individual client optimizes its local model using the loss function defined in (2). Subsequently, the server updates the multi-level global prototypes and soft labels, and then distributes them to clients for the next round of training.

Algorithm 1 FedMPS Framework

Input: number of communication rounds R, number of global epochs T, number of local epochs E, number of clients N

Output: the final model of each client

Server executes:

Initialize multilevel global prototypes $\bar{p}^{level} \leftarrow \{\}$
Initialize global soft labels $\bar{q} \leftarrow \{\}$
for $r = 1, 2, \ldots, R$ **do**
  for $i = 1, 2, \ldots, N$ **in parallel do**
    $p_i^{level} \leftarrow \text{LocalUpdate}(i, \bar{p}^{level}, \bar{q})$
  end for
  Generate global soft labels $\bar{q} \leftarrow \text{GlobalUpdate}(p^h)$
  Aggregate multilevel global prototype $\bar{p}^{level}$
end for

LocalUpdate$(i, \bar{p}^{level}, \bar{q})$

for $e = 1, 2, \ldots, E$ **do**
  for batch $(x, y) \in D_i$ **do**
    Compute classification loss ℓ_{sup}
    Compute multilevel contrastive loss ℓ_{mcon} by Eq. (4)
    Compute soft label loss ℓ_{soft} by Eq. (6)
    Compute total loss ℓ by Eq. (2)
    Update local model according to the total loss ℓ
  end for
end for
Aggregate multilevel local prototype p_i^{level}
return multilevel local prototype p_i^{level}

GlobalUpdate(p^h)

for $t = 1, 2, \ldots, T$ **do**
  for batch $(p^h, \bar{y}) \in P_i$ **do**
    Compute loss ℓ_{global} by Eq. (7)
    Update global model according to the loss ℓ_{global}
  end for
end for
Compute global soft labels $\bar{q}$ by Eq. (8)
return global soft labels $\bar{q}$

IV. Experiments

This section evaluates the effectiveness of FedMPS across seven key aspects: experimental setup, classification accuracy, communication efficiency, heterogeneous robustness, sensitivity analysis, ablation studies, and privacy preservation analysis.

A. Experimental Setup

1) Datasets: We conduct experiments on six public datasets: 1) Flowers[1] (4242 flower images with five classes); 2) DeFungi [52] (9114 microscopic fungus images with five classes); 3) RealWaste[2] (4753 waste images with nine classes); 4) CIFAR-10 (60 000 universal object images with ten classes); 5) Fashion-MNIST (70 000 clothing images with ten classes); and (6) Femnist[3] (1 37 640 character images with 62 classes). These datasets exhibit diversity in the number of classes, sample sizes, feature granularities, and application domains, ensuring comprehensive evaluation of FedMPS's generalizability.

2) Baselines of FL: Seven state-of-the-art algorithms including FedAvg [9], FedProx [10], MOON [11], FedNTD [43], FedGKD [51], FedProc [12], and FedProto [16] are used as comparisons with FedMPS. FedAvg [9], FedProx [10],

[1]https://www.kaggle.com/datasets/alxmamaev/flowers-recognition
[2]https://archive.ics.uci.edu/dataset/908/realwaste
[3]https://leaf.cmu.edu

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

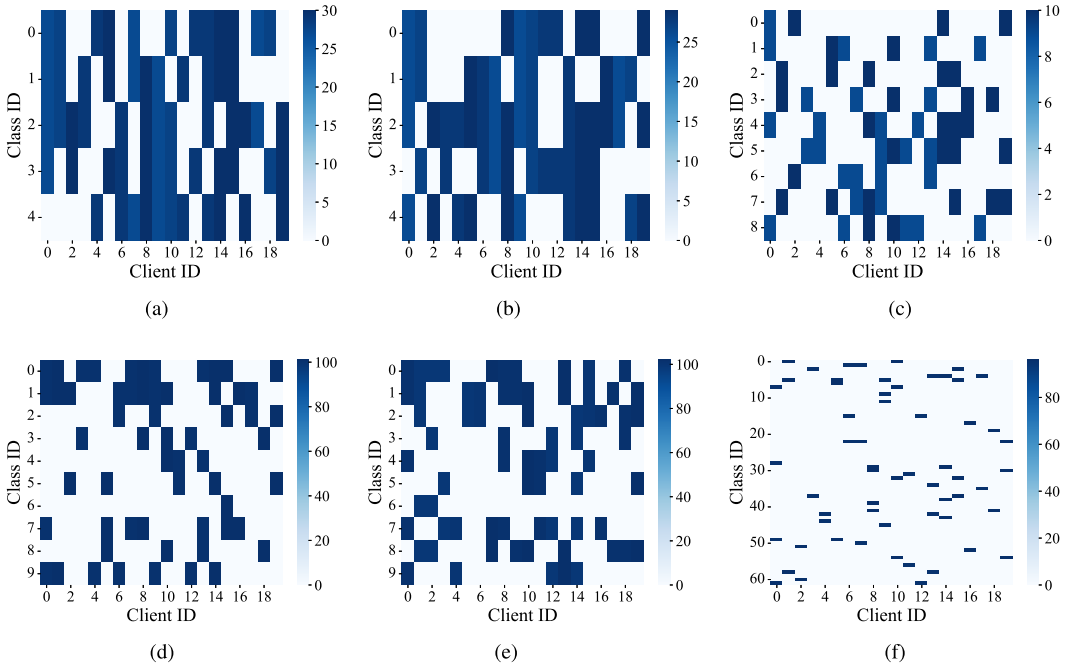


Fig. 3. Data distribution across clients when $n = 3$. The horizontal axis represents the client IDs, while the vertical axis represents the class IDs. Each cell indicates the number of samples belonging to the corresponding class for each client. (a) Flowers. (b) DeFungi. (c) RealWaste. (d) CIFAR-10. (e) Fashion-Mnist. (f) Femnist.

TABLE II
SAMPLING PARAMETERS OF DATASETS

Dataset	classes	$\|A_{ij}\|$	k	t	std
Flowers	5	30	28	10	2
DeFungi	5	29	27	7	2
RealWaste	9	10	9	4	1
CIFAR-10	10	110	100	15	1
Fashion-MNIST	10	110	100	15	2
Femnist	62	96	95	15	1

MOON [11], FedNTD [43], and FedGKD [51] are based on model parameter aggregation, whereas FedProto [16] adopts prototype aggregation. In contrast, FedProc [12] combines both parameter and prototype aggregation. Additionally, FedNTD [43] and FedGKD [51] are gradient-based methods with soft labels. Both MOON [11] and FedProc [12] employ CL to update local models. This explores the comparison between FedMPS and FL approaches based on parameter aggregation, prototype aggregation, soft labels, and CL.

3) Network Architectures: We follow the network of MOON [11] for CIFAR-10, and this is also used for RealWaste, Flowers, and DeFungi. For Femnist and Fashion-MNIST, we follow the network as FedProto [16]. Next, we divide the entire network into a low-level encoder, a high-level encoder, and an output layer. For CIFAR-10, RealWaste, Flowers, and DeFungi, the low-level encoder is a CNN with two 5×5 convolutional layers, followed by a 2×2 max-pooling operation. This outputs normalized 2-D embeddings as low-level representations. The high-level encoder consists of one fully connected layer with 120 units activated by ReLU, producing normalized high-level embeddings. The output layer contains two fully connected layers with ReLU activation (the first with 84 units and the second with class number units). For Femnist, the low-level encoder and the high-level encoder are similar to CIFAR-10 except for the channels and kernel size, and the output layer only uses one fully connected layer. For Fashion-Mnist, the low-level encoder and the high-level encoder both have only one convolution pooling layer, and the output layer is only one fully connected layer. All baselines share identical architectures with FedMPS to ensure fair comparison.

4) Data Partitions: As described in FedProto [16], we partition non-i.i.d. data among clients following a few-shot learning setting, aiming to evaluate the applicability of FL under limited sample conditions. Specifically, each client collects data through n-way k-shots sampling, where n is the number of classes and k is the number of samples per class. Defining n and k as their average values, a random noise denoted as std is added to each client's values. This ensures that the classes owned by each client are not strictly consistent and may overlap, while the number of samples per class also varies across clients. For class j, its train set A_j is divided into N subsets A_{ij} for client i. From each A_{ij}, k samples are randomly selected for training, where $k <= |A_{ij}| <= |A_j|/N$. For the test set of client i, we randomly select a subset from the test dataset with a fixed number t of samples for each class assigned to client i. To ensure a fair comparison, we initially set $n = 3$ to establish a baseline consistent with FedProto [16]. TABLE II presents the sampling parameters, which are determined based on the characteristics of each dataset and aligned with the setup in FedProto [16]. Fig. 3 shows the data distributions across 20 clients on six datasets when $n = 3$. It can be observed that this partitioning approach ensures a non-i.i.d. data distribution for each client. As the number of classes in the dataset increases, such as in Femnist, the probability of

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
TOP-1 TEST ACCURACY OF ALL METHODS ON SIX DATASETS

Method	Flowers	DeFungi	RealWaste	CIFAR-10	Fashion	Femnist
FeAvg [9]	55.46±0.07*	65.02±2.22*	49.90±1.92*	60.91±1.99*	84.21±3.39*	67.64±2.57*
FedProx [10]	56.89±0.39*	65.79±2.84*	50.14±2.34*	61.33±2.09*	84.12±3.39*	67.65±2.19*
MOON [11]	57.93±2.00*	64.48±1.84*	50.35±1.69*	58.14±2.11*	83.49±4.39*	66.83±3.04*
FedNTD [43]	55.67±1.16*	64.74±2.35*	57.01±1.43*	57.45±1.36*	84.26±3.42*	68.59±0.77*
FedGKD [51]	57.50±1.19*	66.19±2.76*	56.15±2.21*	60.37±1.22*	87.01±2.27*	71.08±0.94*
FedProc [12]	55.00±1.27*	62.98±2.97*	47.88±0.80*	57.41±2.17*	81.89±4.24*	64.93±2.14*
FedProto [16]	65.01±1.60*	70.81±0.85	69.24±4.02	75.10±0.94*	93.07±0.56	94.89±1.13
FedMPS	**68.55±0.44**	**74.44±1.78**	**72.99±3.02**	**80.14±0.30**	**93.58±0.89**	**95.57±1.37**

*We run three trials and report the mean and standard deviation. * indicates a significant difference between the method and the proposed method, with the p-value less than 0.05, based on the Wilcoxon rank-sum test.*

class overlap between clients decreases, thereby enhancing the heterogeneity of the data.

5) Implementation Details: PyTorch is employed to implement FedMPS and other baselines, with experiments conducted on an NVIDIA GeForce RTX 3070 GPU running Windows 10. The number of clients N is set to 20. The SGD optimizer is used with a momentum of 0.5 and a default learning rate of 0.01, except for FedGKD [51], which uses a learning rate of 0.05 on RealWaste, CIFAR-10, Fashion-MNIST, and Femnist. The local batch size is set to 8 for all approaches, and the global batch size is set to 4 for FedMPS. The local epoch is set to 1 across all approaches, except for FedNTD [43], which adopts 5 local epochs on RealWaste, CIFAR-10, and Fashion-MNIST. The global epoch is set to 6 for FedMPS. The temperature parameter used in MOON [11], FedProc [12], and FedMPS is set to 0.5.

B. Accuracy Results

For FedMPS, we set the weight λ of the multi-level prototype contrastive loss and the weight β of the high-level contrastive loss to 1, treating them as equally important as the classification loss to simplify optimization. Subsequently, we tune the weight of the low-level contrastive loss α from {0.001, 0.01, 0.2, 0.4, 0.6, 0.8, 1} and the weight of its soft label loss μ from {0.01, 0.1, 1, 5, 10}. The optimal α values for Flowers, DeFungi, RealWaste, CIFAR-10, Fashion-MNIST, and Femnist are 1, 1, 1, 0.2, 1, and 1, respectively, while the optimal μ values are 1, 0.01, 0.01, 5, 1, and 10. Note that both FedProx [10] and MOON [11] have the weight μ of the loss term as a hyperparameter. In FedProx [10], μ (the weight of proximal loss term) is selected in the range of {0.001, 0.01, 0.1, 1}, commonly employed in prior studies [10]. The optimal μ values for FedProx [10] are 0.01, 0.1, 0.001, 0.01, 0.01, and 0.1 across the six datasets. In MOON [11], μ (the weight of model-contrastive loss term) is tuned from the range {0.1, 1, 5, 10} [11]. The optimal μ values for MOON [11] are 0.1, 0.1, 0.1, 5, 1, and 1, respectively. Unless stated otherwise, we use these weight settings for all subsequent experiments.

Table III presents the top-1 test accuracy of all approaches under the default settings when $n = 3$, based on three independent runs. We apply the widely used Wilcoxon rank-sum test [53] to conduct the significance tests for each pair of algorithms. For prototype-based approaches, including FedProto [16] and FedMPS, we report the label predictions with the smallest distance between the high-level features and global prototypes of each class, following FedProto [16]. Other methods all rely on the transmission of model parameters, enabling prediction based on class probabilities. It is observed that FedProx [10] with a proximal term, MOON [11] with model-CL, and both FedNTD [43] and FedGKD [51] with knowledge distillation achieve limited performance improvements over FedAvg [9] in most scenarios. The CL module in FedProc [12] with all class prototypes may lead to overfitting of the local model, thereby degrading performance. The performance of FedProto [16] and FedMPS utilizing prototype knowledge is improved by 15.38% on average compared with approaches based on model parameter knowledge, highlighting the effectiveness of prototype-based methods in non-i.i.d. settings. In particular, FedMPS outperforms FedNTD [43] and FedGKD [51] by 16.26% and 14.50%, respectively, underscoring the advantage of prototype-based soft labels over parameter-based ones. Compared with other methods, FedMPS consistently demonstrates superior performance across various datasets. The Wilcoxon rank-sum test reveals statistically significant differences between FedMPS and most of the compared methods across nearly all datasets. Although FedProto [16] is not significantly different from FedMPS on DeFungi, RealWaste, Fashion, and Femnist, the effect sizes (Cliff's Delta) are 0.89, 0.56, 0.11, and 0.33, respectively, suggesting FedMPS has a large effect size difference from FedProto [16] on DeFungi and RealWaste. It outperforms FedProto [16] (the

TABLE IV
TOP-1 TEST ACCURACY ON IMAGENET UNDER THREE INDEPENDENT RUNS WITH $n = 3$ AND $N = 20$

Method	FedProto [16]	FedMPS
ImageNet	98.55±0.59	**98.89±0.20**

TABLE V
COMMUNICATION PARAMETERS

Method	Comm medium	Comm params
FedAvg [9]	gradients	1,240,120
FedProx [10]	gradients	1,240,120
MOON [11]	gradients	1,240,120
FedNTD [43]	gradients	1,240,120
FedGKD [51]	gradients	2,480,240
FedProc [12]	gradients+prototypes	1,247,200
FedProto [16]	prototypes	7,080
FedMPS	multi-level prototypes +soft labels (one-way)	30,780

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

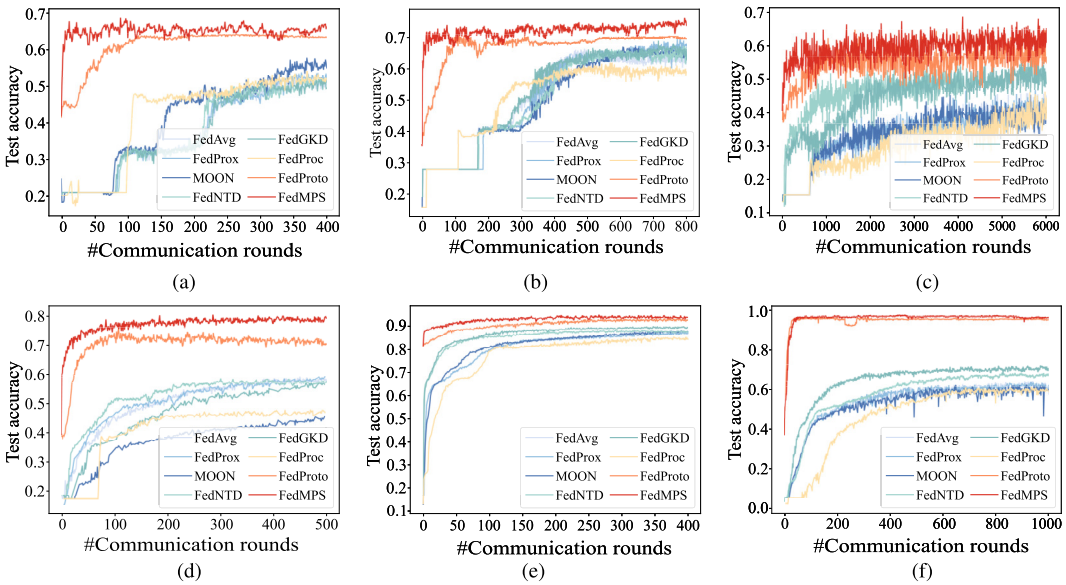


Fig. 4. Test accuracy across different communication rounds when $n = 3$. (a) Flowers. (b) DeFungi. (c) RealWaste. (d) CIFAR-10. (e) Fashion-Mnist. (f) Femnist.

second-best approach) by 2.86% in classification accuracy on average across all datasets. Specifically, on the three-channel image datasets, namely Flowers, DeFungi, RealWaste, and CIFAR-10, the average excess of FedProto [16] is 3.99%. On the one-channel gray-scale image datasets, namely Fashion-MNIST and Femnist, the performance improvement is less pronounced. This is because the multi-level feature-prototype contrastive module of FedMPS incorporates both semantic and detailed information, such as color and texture, compared to previous methods. We further evaluate the applicability of FedMPS on the ImageNet[4] dataset, which features a large number of classes (1000) and high-resolution images (224 × 224), using ResNet-18 as the backbone. As shown in Table IV , FedMPS outperforms FedProto [16] with lower variance, validating its stability under more challenging scenarios.

C. Communication Efficiency

Fig. 4 shows the test accuracy at each training round. For clarity, we use the same number of rounds for all algorithms, but the results in Table III are still the results of each algorithm after its respective convergence round. As observed, FedMPS requires fewer rounds to converge than other algorithms, demonstrating its faster convergence rate. Specifically, for Flowers and DeFungi, the test accuracy curves of FedAvg [9], FedProx [10], MOON [11], FedNTD [43], FedGKD [51], and FedProc [12] converge slowly and show a tendency to flatten out and rise again over a period of rounds. This phenomenon is likely caused by the uniform assignment of aggregated parameters to each client, which can lead to deviations from their intrinsic data distributions and reduce the effectiveness of these methods in mitigating data heterogeneity. Furthermore, for a fixed test accuracy value on the y-axis, FedMPS consistently reaches the threshold with the fewest training rounds.

[4]https://image-net.org/

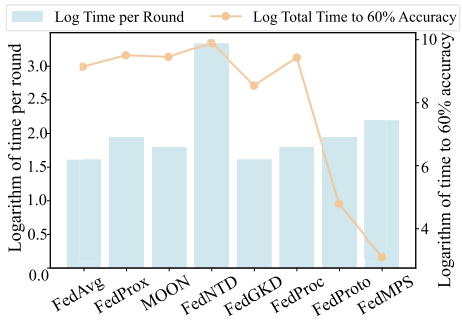


Fig. 5. Average time consumption per round and total time consumption to 60% accuracy on CIFAR-10 when $n = 3$.

In addition, communication cost is an important consideration in FL, where network bandwidth constraints and an increase in the number of clients may lead to client dropouts. Table V presents the number of parameters communicated per round. As shown in the table, while FedMPS involves slightly more communication parameters than FedProto [16], it transmits only 1/40 of the parameters compared to most other algorithms. Therefore, FedMPS achieves significant performance improvements while effectively controlling communication costs.

Fig. 5 illustrates the average time consumption of training per round and total time to 60% accuracy on the CIFAR-10 dataset with $n = 3$. FedAvg [9] has the shortest runtime per round due to its straightforward algorithmic structure. FedMPS requires the second-longest but acceptable runtime per round compared to other methods, primarily due to its comprehensive optimization procedures. Despite its longer runtime per round, FedMPS achieves the shortest total time to reach 60% accuracy due to its faster convergence and superior performance. Therefore, the tradeoff between improved performance and longer training time within an acceptable range is justified. In terms of performance, FedMPS consistently achieves higher accuracy

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VI
AVERAGE TEST ACCURACY UNDER VARYING THE AVERAGE NUMBER OF CLASSES n

n	Method	Flowers	DeFungi	RealWaste	CIFAR-10	Fashion	Femnist
$n=3$	FeAvg [9]	55.38±12.10	66.91±11.13	47.19±15.16	63.50±14.80	88.94±7.82	64.11±17.84
	FedProx [10]	57.33±12.93	69.70±12.43	46.88±20.09	63.60±14.80	88.87±7.77	64.78±18.84
	MOON [11]	59.63±11.24	67.02±13.71	48.54±21.18	57.16±19.83	89.64±7.50	63.64±23.70
	FedNTD [43]	57.25±9.05	67.38±14.17	58.23±18.41	58.78±15.82	88.49±7.44	68.22±17.66
	FedGKD [51]	59.04±10.94	68.99±15.70	59.17±19.61	62.06±13.52	89.91±8.16	71.97±12.02
	FedProc [12]	54.04±12.90	61.79±15.59	46.88±18.78	58.42±19.76	87.87±8.38	62.44±23.27
	FedProto [16]	64.17±15.84	70.54±15.39	64.58±22.88	75.67±12.86	93.42±4.96	96.33±4.26
	FedMPS	**68.58±15.41**	**76.07±14.15**	**69.90±17.37**	**80.31±8.91**	**94.76±5.74**	**97.31±3.58**
$n=4$	FeAvg [9]	54.33±12.14	66.96±15.73	49.67±12.33	63.50±8.50	84.58±12.76	67.33±15.64
	FedProx [10]	55.50±15.76	69.35±14.80	52.38±13.14	63.20±8.90	84.68±12.58	66.22±16.98
	MOON [11]	57.38±13.73	69.05±11.50	53.35±14.55	60.69±10.94	85.11±12.50	66.54±16.60
	FedNTD [43]	55.92±10.92	67.92±15.36	51.67±13.61	62.80±10.40	85.38±10.84	71.44±15.05
	FedGKD [51]	57.54±13.65	69.23±15.32	50.40±12.40	63.05±11.10	86.52±11.55	73.48±14.10
	FedProc [12]	52.33±10.84	61.96±16.14	51.96±15.23	60.69±11.24	83.23±12.73	66.57±15.40
	FedProto [16]	64.54±16.45	70.42±17.59	56.56±17.59	62.89±12.74	89.50±8.87	91.97±7.13
	FedMPS	**66.79±13.96**	**77.20±13.23**	**63.92±13.59**	**72.10±9.12**	**90.51±8.19**	**92.27±7.41**
$n=5$	FeAvg [9]	53.04±7.87	63.16±10.97	50.79±11.90	68.00±6.80	88.01±7.76	69.99±14.37
	FedProx [10]	53.08±9.45	61.55±10.30	50.69±14.59	68.80±8.00	87.90±8.20	69.59±14.19
	MOON [11]	54.46±9.68	61.70±11.92	49.17±13.31	66.11±7.27	88.22±8.06	69.75±13.42
	FedNTD [43]	53.17±7.36	**66.61±10.86**	**55.46±10.70**	**68.84±6.30**	87.36±7.25	71.84±12.09
	FedGKD [51]	53.75±7.99	62.50±10.73	54.31±10.71	67.41±7.95	88.60±7.94	76.43±12.28
	FedProc [12]	52.29±11.23	58.81±11.12	51.38±16.74	65.58±8.52	85.34±6.60	70.79±13.53
	FedProto [16]	56.42±11.55	60.18±13.45	48.00±15.60	53.67±10.21	88.64±6.14	91.74±6.35
	FedMPS	**59.25±14.47**	64.94±10.39	55.02±13.06	66.27±8.60	**89.49±6.11**	**92.62±5.56**

We report the mean and standard deviation of test accuracies among all clients under the same random seed, which represent the average and variation in performance across clients.

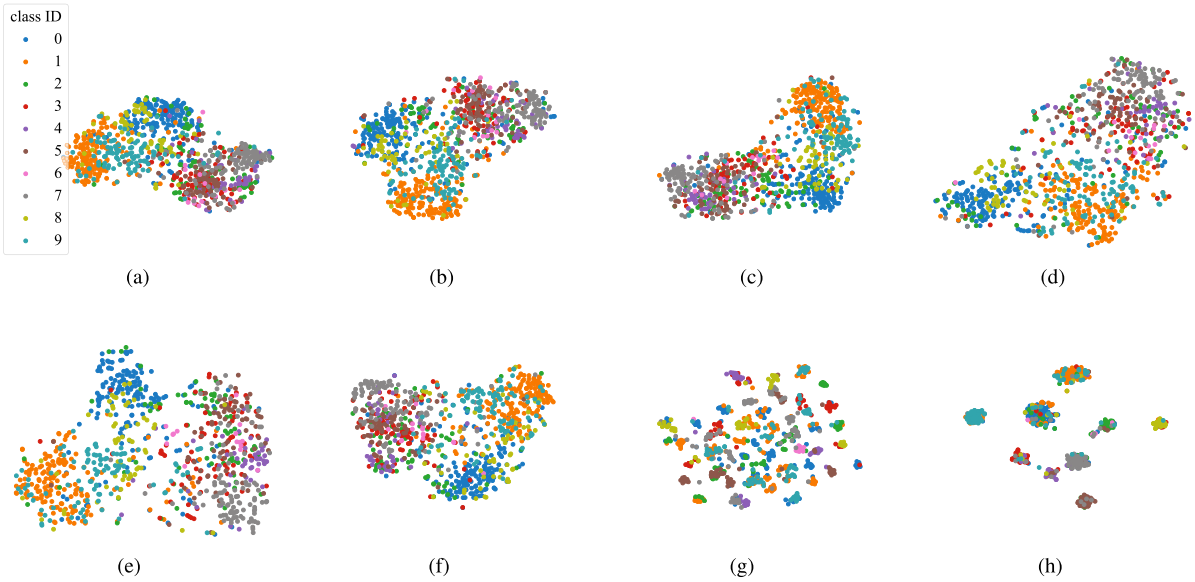


Fig. 6. T-SNE visualizations of FedMPS and other approaches on the CIFAR-10 test set with n = 3, N = 20. Each point represents a hidden vector of a data sample, with colors indicating different classes. The CIFAR-10 dataset consists of ten classes that are distributed among all clients. The more clearly the colors are separated across classes, the better the model has learned class-discriminative features. (a) FedAvg. (b) FedProx. (c) MOON. (d) FedNTD. (e) FedGKD. (f) FedProc. (g) FedProto. (h) FedMPS.

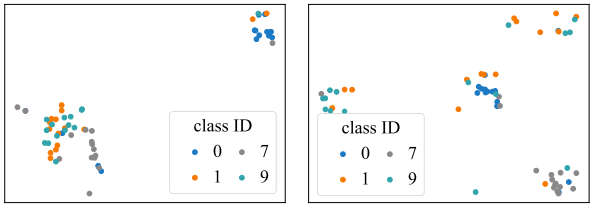
than other algorithms, making it particularly advantageous in scenarios where high-performance models are essential. On the other hand, due to its faster convergence, FedMPS requires fewer rounds and less total time to achieve the same level of performance as other algorithms. This makes it an excellent choice for scenarios prioritizing training efficiency, such as resource-constrained environments. Therefore, FedMPS is well-suited for both high-performance and high-efficiency scenarios.

D. Visualization Comparisons

T-SNE [54] is a technique for visualizing hidden representations in a 2-D space to illustrate feature discrimination. Fig. 6 shows the visualization results of all approaches on the CIFAR-10 test set, in which points of different colors represent sample embeddings belonging to distinct classes. It can be seen in Fig. 6(h) that the samples within each class cluster are more concentrated, and the separation between classes is more distinct. These results demonstrate that FedMPS

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) (b)

Fig. 7. T-SNE visualizations of FedProto [16] and FedMPS on the CIFAR-10 test set for the client C_1 with $n = 3$, $N = 20$. The CIFAR-10 dataset consists of four classes distributed across the client C_1, highlighting the class-discriminative capability within a single client. (a) FedProto. (b) FedMPS.

TABLE VII
ABLATION STUDIES ON LOSS COMPONENTS, REPORTING THE MEAN AND STANDARD DEVIATION OF TEST ACCURACIES ACROSS CLIENTS

No.	ℓ^l_{con}	ℓ^h_{con}	ℓ_{soft}	Acc
(1)	0	0	0	74.67±12.32
(2)	1	0	0	76.75±12.46
(3)	0	1	0	79.06±8.72
(4)	0	0	1	75.10±12.33
(5)	1	1	0	79.69±10.25
(6)	1	1	1	80.31±8.91

exhibits a superior ability in representation learning, enabling more effective differentiation between samples from different classes. Furthermore, to evaluate the representation learning capability of a single local model, we visualize the representations of the local data from the first client model for both FedProto [16] and FedMPS. In Fig. 7(a), we can see that FedProto [16] primarily forms two clusters, despite the presence of four classes in the first client's dataset. In contrast, FedMPS effectively separates the local samples into four distinct clusters, demonstrating a significant advantage in local representation learning. These visualization results confirm that our method significantly enhances representation learning in FL under data heterogeneity.

E. Heterogeneous Robustness

We investigate the heterogeneous robustness of all approaches by varying the average number of classes n, the random noise std, and the number of clients N on CIFAR-10, respectively.

As the average number of classes (n) increases, data diversity is greater, leading to an increase in classification difficulty. Following the approach in FedProto [16], we further set n to 4 and 5 to evaluate the robustness of our method under different levels of heterogeneity. The average and standard deviation results among clients are shown in Table VI. As observed, FedMPS consistently achieves the best performance and low differences among clients across different values of n in most tasks, surpassing the second-best algorithm by an average of 3.54%. Although it does not achieve the highest accuracy on DeFungi, RealWaste, and CIFAR-10 when $n = 5$, its performance remains comparable to that of the best algorithm. Moreover, it demonstrates small performance variance across clients, indicating its effectiveness in mitigating performance discrepancies caused by data heterogeneity.

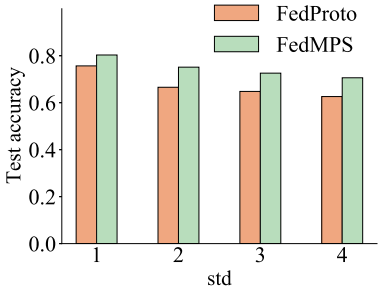


Fig. 8. Average test accuracy on CIFAR-10 under varying the random noise std.

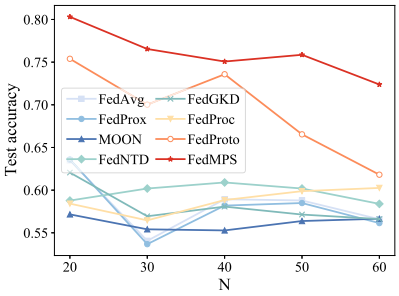


Fig. 9. Average test accuracy on CIFAR-10 under varying the number of clients N.

As the random noise (std) increases, the disparity between the number of classes and the number of samples in each class owned by each client also increases, resulting in a higher degree of data heterogeneity among clients. Fig. 8 compares the performance of FedMPS and FedProto [16] on CIFAR-10 with $n = 3$ under different values of std. It is evident that as std increases, the performance of both FedMPS and FedProto [16] declines. However, FedMPS consistently outperforms FedProto [16] and exhibits a smaller rate of decline.

As the number of clients (N) grows, each client interacts with an increasing number of other clients that have heterogeneous data, heightening the requirement for collaborative learning. Therefore, we evaluate performance with a larger number of clients on the CIFAR-10 dataset with $n = 3$, specifically setting N to 30, 40, 50, and 60. The results are shown in Fig. 9, demonstrating that FedMPS continues to outperform other algorithms even with a larger number of clients. Moreover, its performance declines at a slower rate compared to FedProto [16] as N increases from 40 to 60.

F. Sensitivity Analysis

In this section, we conduct a sensitivity analysis of two temperature coefficients, τ_1 and τ_2, to evaluate their impacts on model performance.

The coefficient τ_1 is the temperature parameter of the multi-level prototype CL loss in (3), which regulates the class probability distribution and influences the model's focus on negative samples. Following MOON [11], we tune over {0.1, 0.5, 1.0}. The optimal value of τ_1 for the RealWaste and other datasets is 1.0 and 0.5, respectively. Observed in Fig. 10(a), variations in τ_1 do not lead to significant performance changes across all datasets. The standard deviations in performance across different values of τ_1 for Flowers, DeFungi, RealWaste,

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

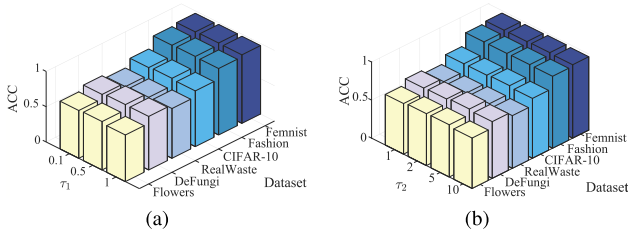


Fig. 10. Test accuracy of FedMPS trained with different temperature coefficients. (a) Effect of τ_1. (b) Effect of τ_2.

CIFAR-10, Fashion-MNIST, and Femnist are 1.29%, 1.14%, 1.01%, 0.77%, 0.36%, and 0.40%, respectively. This indicates that FedMPS is relatively insensitive to the choice of τ_1.

The coefficient τ_2 is the temperature parameter used for soft label generation in (5) and (8), which controls the smoothness of the output probability distribution. When $\tau_2 > 1$, the logits become smoother, allowing the model to learn from class probabilities beyond the highest-confidence predictions. Following [55], we explore values from {1, 2, 5, 10}. The optimal τ_2 across all datasets is consistently 5. The results are depicted in Fig. 10(b), showing that performance remains stable despite variations in τ_2. The standard deviations across different values of τ_2 for Flowers, DeFungi, RealWaste, CIFAR-10, Fashion-MNIST, and Femnist are 1.20%, 1.04%, 0.49%, 0.73%, 0.13%, and 0.37%, respectively. Notably, the model demonstrates even lower sensitivity to τ_2 compared to τ_1, suggesting that FedMPS exhibits robustness to variations in the temperature coefficient τ_2 for soft labels.

The results highlight that FedMPS maintains stable performance across different temperature settings. The low standard deviations across all datasets suggest that FedMPS does not heavily depend on precise temperature tuning, making it adaptable to diverse scenarios without extensive hyperparameter optimization.

G. Ablation Experiments

We conduct ablation studies on FedMPS using CIFAR-10 with $n = 3$ to examine the impact of three loss components in (2). The results are reported in Table VII. All of them use the classification loss ℓ_{sup} as the baseline. It is observed from No. (2) that when the low-level CL loss ℓ^l_{con} is added to the baseline No. (1), the performance is improved by 2.08%. This confirms that ℓ^l_{con} provides additional detailed information for better learning. Furthermore, No. (3) demonstrates that adding the high-level contrastive loss ℓ^h_{con} to the baseline results in a 4.39% increase in average accuracy while significantly reducing performance variance across clients. This highlights the crucial role of incorporating the whole semantic information to enhance model performance. Additionally, the soft label loss ℓ_{soft} also contributes to performance improvements, yielding a 0.62% increase corresponding to No. (5) and No. (6). As evidenced by the optimal weight $\mu = 10$ for ℓ_{soft} on Femnist, datasets with a large number of classes require a higher weight for ℓ_{soft} to facilitate learning more informative class relationships. These ablation studies demonstrate the effectiveness of enhancing information diversity from multiple perspectives in facilitating collaborative learning across clients.

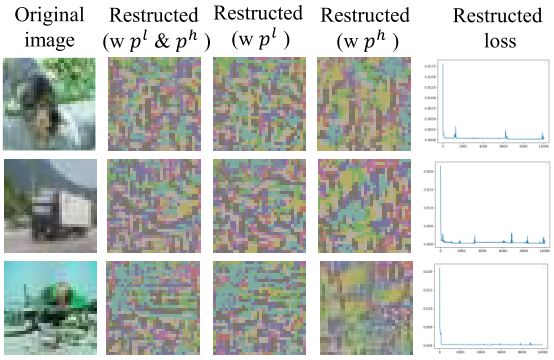


Fig. 11. Results of the feature space hijacking attack on FedMPS (Client 1, CIFAR-10, $n = 3$). From left to right: 1) original images; 2) reconstructed images using both low- and high-level prototypes; 3) reconstructed images using only low-level prototypes; 4) reconstructed images using only high-level prototypes; and 5) the reconstruction loss curve over 10 000 training steps.

H. Privacy Preservation Analysis

Following MPFT [56], we adopt the Feature Space Hijacking Attack to empirically evaluate the privacy risk of FedMPS. This attack operates in a opaque-box setting [57], where the adversary can query the client model to obtain intermediate outputs and has access to the uploaded multi-level prototypes for input reconstruction.

Fig. 11 illustrates the reconstruction results of images based on prototypes. We observe that the attacker consistently fails to reconstruct recognizable images, regardless of whether single-level or multi-level prototypes are used. This suggests that FedMPS maintains privacy-preserving abilities under feature-based attacks.

V. Conclusion

In this study, we propose a novel FL framework (FedMPS) that integrates multi-level prototype-based CL and soft label generation to enhance performance while controlling communication overhead under data heterogeneity in FL. FedMPS introduces a method to extract valuable information from various scales and latent knowledge across classes by utilizing multi-level local feature-global prototype CL and soft label interactions, which provides new insights into the synergy of diverse information from prototypes in FL. Extensive experiments demonstrate that FedMPS significantly outperforms existing methods in few-shot federated image classification tasks.

Moreover, a promising future direction is to extend FedMPS to vertical FL (VFL), where a key challenge lies in feature misalignment. Another crucial avenue is adapting FedMPS to dynamic data, such as time series, where sequential dependencies and pattern variability pose fundamental challenges for effective temporal knowledge alignment.

References

[1] J. Wang, A. Pal, Q. Yang, K. Kant, K. Zhu, and S. Guo, "Collaborative machine learning: Schemes, robustness, and privacy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 9625–9642, Dec. 2023.

[2] J. O. du Terrail et al., "Federated learning for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer," *Nat. Med.*, vol. 29, no. 1, pp. 135–146, Jan. 2023.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[3] T. Tang et al., "Personalized federated graph learning on non-IID electronic health records," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 11843–11856, Sep. 2024.

[4] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020.

[5] J. Chen, J. Xue, Y. Wang, L. Huang, T. Baker, and Z. Zhou, "Privacy-preserving and traceable federated learning for data sharing in industrial IoT applications," *Expert Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119036.

[6] B. Xue, Y. He, F. Jing, Y. Ren, L. Jiao, and Y. Huang, "Robot target recognition using deep federated learning," *Int. J. Intell. Syst.*, vol. 36, no. 12, pp. 7754–7769, Dec. 2021.

[7] D. Cheng, L. Zhang, C. Bu, X. Wang, H. Wu, and A. Song, "ProtoHAR: Prototype guided personalized federated learning for human activity recognition," *IEEE J. Biomed. Health Informat.*, vol. 27, no. 8, pp. 3900–3911, Aug. 2023.

[8] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel, "Federated generalized face presentation attack detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 103–116, Jan. 2024.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. 3rd Mach. Learn. Syst. Conf.*, vol. 2, 2020, pp. 429–450.

[11] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10713–10722.

[12] X. Mu et al., "FedProc: Prototypical contrastive federated learning on non-IID data," *Future Gener. Comput. Syst.*, vol. 143, pp. 93–104, Jun. 2023.

[13] B. Wei, J. Li, Y. Liu, and W. Wang, "Non-IID federated learning with sharper risk bound," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 5, pp. 6906–6917, May 2024.

[14] U. Michieli and M. Ozay, "Prototype guided federated learning of visual feature representations," 2021, *arXiv:2105.08982*.

[15] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-iid data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 5972–5984.

[16] Y. Tan et al., "FedProto: Federated prototype learning across heterogeneous clients," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, 2022, pp. 8432–8440.

[17] B. Yan, H. Zhang, M. Xu, D. Yu, and X. Cheng, "FedRFQ: Prototype-based federated learning with reduced redundancy, minimal failure, and enhanced quality," *IEEE Trans. Comput.*, vol. 73, no. 4, pp. 1086–1098, Apr. 2024.

[18] Y. Qiao et al., "MP-FedCL: Multiprototype federated contrastive learning for edge intelligence," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 8604–8623, Mar. 2024.

[19] Y. Qiao, M. S. Munir, A. Adhikary, A. D. Raha, S. H. Hong, and C. S. Hong, "A framework for multi-prototype based federated learning: Towards the edge intelligence," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2023, pp. 134–139.

[20] S. Guo, H. Wang, and X. Geng, "Dynamic heterogeneous federated learning with multi-level prototypes," *Pattern Recognit.*, vol. 153, Sep. 2024, Art. no. 110542.

[21] T. Gao, X. Liu, Y. Yang, and G. Wang, "FEDMBP: Multi-branch prototype federated learning on heterogeneous data," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2023, pp. 2180–2184.

[22] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "LayerCAM: Exploring hierarchical class activation maps for localization," *IEEE Trans. Image Process.*, vol. 30, pp. 5875–5888, 2021.

[23] S. K. Reed, "Pattern recognition and categorization," *Cogn. Psychol.*, vol. 3, no. 3, pp. 382–407, Jul. 1972.

[24] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4080–4090.

[25] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 8334–8343.

[26] J. Liu, Y. Bao, G. S. Xie, H. Xiong, J. J. Sonke, and E. Gavves, "Dynamic prototype convolution network for few-shot semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11553–11562.

[27] Q. Wang, D. Zhou, Y. Zhang, D. Zhan, and H. Ye, "Few-shot class-incremental learning via training-free prototype calibration," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 15060–15076.

[28] Y. Xiao, Y. Jin, and K. Hao, "Adaptive prototypical networks with label words and joint representation learning for few-shot relation classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 3, pp. 1406–1417, Mar. 2023.

[29] Y. Zhao et al., "Personalized federated few-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2534–2544, Feb. 2024.

[30] M. Pang, B. Wang, S. Huang, Y.-M. Cheung, and B. Wen, "A unified framework for bidirectional prototype learning from contaminated faces across heterogeneous domains," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1544–1557, 2022.

[31] L. Zhou, M. Ye, D. Zhang, C. Zhu, and L. Ji, "Prototype-based multisource domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5308–5320, Oct. 2022.

[32] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1735–1742.

[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 1597–1607.

[34] J. Li, P. Zhou, C. Xiong, and S. C. H. Hoi, "Prototypical contrastive learning of unsupervised representations," 2020, *arXiv:2005.04966*.

[35] P. Khosla et al., "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 18661–18673.

[36] J. Cui, Z. Zhong, S. Liu, B. Yu, and J. Jia, "Parametric contrastive learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 715–724.

[37] S. Bae, S. Kim, J. Ko, G. Lee, S. Noh, and S.-Y. Yun, "Self-contrastive learning: Single-viewed supervised contrastive framework using sub-network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 197–205.

[38] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, Jul. 2020, pp. 5132–5143.

[39] D. Alp Emre Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," 2021, *arXiv:2111.04263*.

[40] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu, "FedDC: Federated learning with non-IID data via local drift decoupling and correction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10112–10121.

[41] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "FedAUX: Leveraging unlabeled auxiliary data in federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 5531–5543, Sep. 2023.

[42] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12878–12889.

[43] G. Lee, M. Jeong, Y. Shin, S. Bae, and S.-Y. Yun, "Preservation of the global knowledge by not-true distillation in federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 38461–38474.

[44] J. Kim, G. Kim, and B. Han, "Multi-level branched regularization for federated learning," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, vol. 162, 2022, pp. 11058–11073.

[45] X. Chen, B. Li, and W. Li, "Improve model-contrastive federated learning by momentum contrast," in *Proc. Int. Conf. Mach. Learn. Comput. Appl.*, vol. 12636, 2023, pp. 197–203.

[46] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, *arXiv:1910.03581*.

[47] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, "FedMix: Approximation of mixup under mean augmented federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–19.

[48] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, May 2022, pp. 10174–10183.

[49] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature Commun.*, vol. 13, no. 1, p. 2032, Apr. 2022.

[50] S. Han et al., "FedX: Unsupervised federated learning with cross knowledge distillation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 691–707.

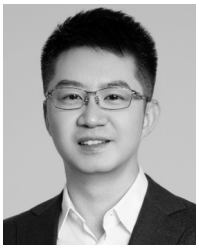
Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[51] D. Yao et al., "FedGKD: Toward heterogeneous federated learning via global knowledge distillation," *IEEE Trans. Comput.*, vol. 73, no. 1, pp. 3–17, Jan. 2024.
[52] M. A. V. Álvarez, L. Sopó, C. J. P. Sopo, F. Hajati, and S. Gheisari, "P456 Defungi: Direct mycological examination of microscopic fungi images," *Med. Mycol.*, vol. 60, Sep. 2022, Art. no. myac072P456.
[53] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Ann. Math. Statist.*, vol. 18, no. 1, pp. 50–60, Mar. 1947.
[54] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[55] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[56] J. Zhang, Y. Duan, S. Niu, Y. Cao, and W. Y. B. Lim, "Enhancing federated domain adaptation with multi-domain prototype-based federated fine-tuning," 2024, *arXiv:2410.07738*.
[57] Z. He, T. Zhang, and R. B. Lee, "Attacking and protecting data privacy in edge–cloud collaborative inference systems," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9706–9716, Jun. 2021.

Wenxin Yang received the B.S. degree from the School of Maritime Economics and Management, Dalian Maritime University, Dalian, China, in 2022. She is currently pursuing the Ph.D. degree with the Department of Systems Engineering, National University of Defense Technology, Changsha, China.

Her current research interests include federated learning.



Xingchen Hu received the B.E. degree in spacecraft design and engineering from Beihang University, Beijing, China, in 2011, the M.E. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2013, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada, in 2017.

He is currently an Associate Professor with the Laboratory for Big Data and Decision, College of Systems Engineering, National University of Defense Technology. His research interests include computational intelligence, granular computing, knowledge discovery and data mining, and evolutionary optimization.

Xiubin Zhu received the B.S. degree in computer science and technology from Xi'an Shiyou University, Xi'an, China, in 2004, and the M.S. and Ph.D. degrees in computer software and theory, control theory, and control engineering from Xidian University, Xi'an, in 2007 and 2018, respectively.

He is currently an Associate Professor with the School of Electro-Mechanical Engineering, Xidian University. His research interests include granular computing, data mining, and pattern recognition.

Rouwan Wu received the Bachelor's degree in engineering from Central South University of Forestry and Technology, Changsha, China, in 2019, and the Master of Engineering degree from the University of Chinese Academy of Sciences, Beijing, China, in 2022. She is currently pursuing the Ph.D. degree with the National University of Defense Technology, Changsha.

She is interested in 3-D reconstruction, NeRF, and localization.

Witold Pedrycz (Life Fellow, IEEE) received the M.Sc., Ph.D., and D.Sci. degrees in automatic control and computer science from the Silesian University of Technology, Gliwice, Poland, in 1977, 1980, and 1984, respectively.

He is currently a Professor and a Canada Research Chair of Computational Intelligence with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland. He has also authored 15 research monographs covering various aspects of computational intelligence, data mining, and software engineering. He has authored or co-authored numerous articles in these areas. His main research directions involve computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering.

Dr. Pedrycz was elected as a Foreign Member of the Polish Academy of Sciences in 2009. He is intensively involved in editorial activities. He is a fellow of the Royal Society of Canada. He was a recipient of the IEEE Canada Computer Engineering Medal, the Cajastur Prize for Soft Computing from the European Centre for Soft Computing, the Killam Prize, and the Fuzzy Pioneer Award from the IEEE Computational Intelligence Society.

Xinwang Liu (Senior Member, IEEE) received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2013.

He is now a Professor at the School of Computer, NUDT. He has published more than 70 peer-reviewed articles, including those in highly regarded journals and conferences such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (T-KDE), IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), IEEE TRANSACTIONS ON MULTIMEDIA (T-MM), IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (T-IFS), ICML, NeurIPS, CVPR, ICCV, AAAI, and IJCAI. His current research interests include kernel learning, multiview clustering, and unsupervised feature learning.

Dr. Liu is an Associate Editor of IEEE TNNLS and *Information Fusion Journal*. More information can be found at https://xinwangliu.github.io/

Jincai Huang is a Professor at the College of Systems Engineering, National University of Defense Technology, Changsha, Hunan, China. He has published more than 80 SCI/EI-indexed articles. His main research interests include artificial general intelligence, deep reinforcement learning, and multiagent systems.

Mr. Huang serves as a Director of the Machine Learning Committee of the Chinese Association for Artificial Intelligence (CAAI).

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 20,2025 at 08:37:56 UTC from IEEE Xplore. Restrictions apply.