Hypernetwork Aggregation for Decentralized Personalized Federated Learning

Weishi Li[1] , **Yong Peng**[1*] , **Mengyao Du**[1] , **Fuhui Sun**[2] , **Xiaoyan Wang**[2*] , **Li Shen**[3]
[1]College of Systems Engineering, National University of Defense Technology
[2]Information Technology Service Center of People's Court
[3]School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University
{liweishi1028, yongpeng, dumengyao}@nudt.edu.cn,
{sunfh6732,428163395}@163.com, shenli6@mail.sysu.edu.cn

Abstract

Personalized Federated Learning (PFL) meets each user's personalized needs while still facing the high communication costs due to the large amount of data transmission and frequent communication. Decentralized PFL (DPFL) as an alternative discards the central server in PFL, which reduces the pressure of communication and the risk of server failure by using peer-to-peer communication.Nevertheless, DPFL still suffers from the significant communication pressure due to the transmission of a large number of model parameters, especially numerous nodes. To address the issues, we propose a novel personalized framework, DFedHP, in which each client utilizes a hypernetwork to generate the shared part of model parameters and train the personalized parameters separately. The number of parameters in a hypernetwork is much smaller than those in a typical local network, so hypernetwork aggregation reduces communication costs and the risk of privacy leakage. Furthermore, DFedHP can seamlessly integrate into existing DPFL algorithms as a plugin to boost their efficacy. At last, extensive experiments on various data heterogeneous environments demonstrate that DFedHP can reduce communication costs, accelerate convergence rate, and improve generalization performance compared with state-of-the-art (SOTA) baselines.

1 Introduction

Personalized Federated Learning (PFL) [Pillutla *et al.*, 2022] trains customized models for each local client. PFL aims to meet the diverse needs of users and improve the generalization ability and robustness of models in distributed environments [Tan *et al.*, 2022]. To alleviate communication stress and the risks of server failure in centralized federated learning (CFL), DPFL [Li *et al.*, 2023] uses communication topologies to directly exchange model updates with neighbors without a central server [Sabah *et al.*, 2024]. It distributes the communication load to some extent, which is particularly important for large-scale distributed systems.

However, even in DPFL, nodes need to frequently synchronize and update local model parameters or gradient information, increasing communication paths and complexity in the network [Yuan *et al.*, 2024]. Frequent and large data transmissions in such topologies consume a significant amount of network bandwidth, especially on mobile devices or IoT devices (Internet of Things). This not only leads to high communication costs, but may also result in delayed or reduced parameter updates, affecting the consistency and accuracy of the model. Moreover, the large amount of communication increases the time that data is transmitted over the network, providing more potential attack surfaces and the risk of privacy leaks. Recently, researchers propose communication optimization strategies, such as compression [Zhu *et al.*, 2024], optimization algorithms [Wu *et al.*, 2022] and efficient decentralized architectures [Beltrán *et al.*, 2023] to lower communication volume and frequency [Dai *et al.*, 2022]. However, the high communication cost and slow convergence issues have not been fundamentally solved.

Therefore, we propose a new algorithm, the Decentralized Personalized Federated Learning algorithm with Hypernetwork (DFedHP). We deploy hypernetworks on the client-side of the decentralized topologies. The hypernetwork maps input information to the desired target and generates weights of target local model (convolutional networks or recurrent networks) [Ha *et al.*, 2016]. Clients only transmit these smaller hypernet parameters, which greatly reduces the amount of communication data required to transmit a complete model [Chauhan *et al.*, 2024]. This means that hypernetworks decouple communication costs from trainable model sizes. In addition, the clients only need to share the parameters of the hypernetwork, without exposing the complete architecture or weight details of their local models. This approach effectively protects user data privacy by serving as an intermediate abstract layer that prevents direct transmission of sensitive information, providing stronger privacy guarantees. Further, hypernetwork optimization can be performed in a smaller search space (i.e., optimizing the weights of the hypernetwork itself). This not only improves efficiency but also may avoid getting stuck in a local optimum, making the final generated weights more likely to approach an ideal solution [Caldarola *et al.*, 2022]. In sum, clients utilize hypernetworks to generate

*Corresponding authors.

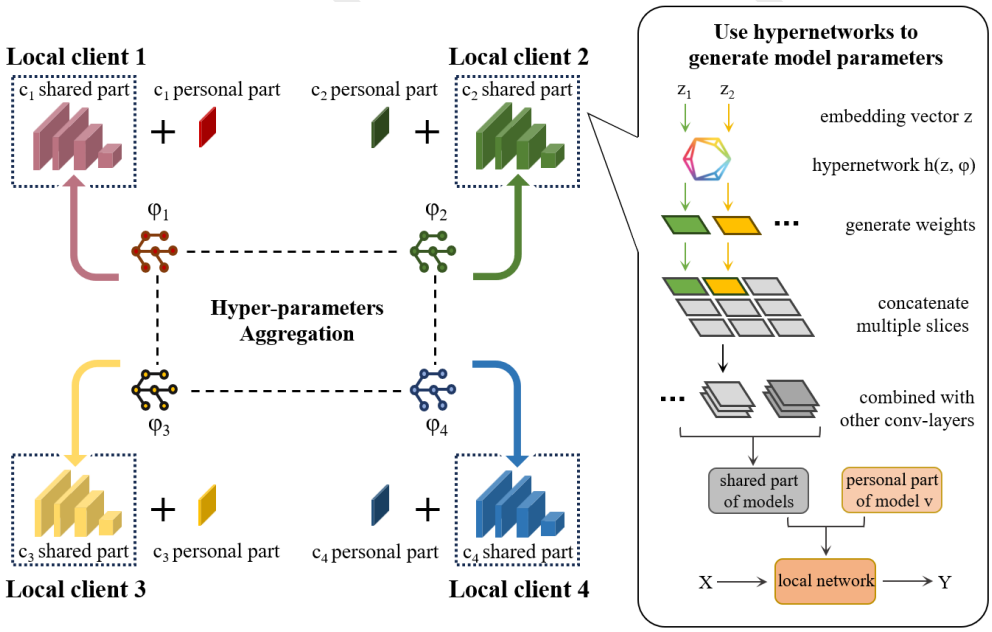

Figure 1: The main pipeline of DFedHP algorithm. Each client unit c_i consists of a personalized part v_i and a hypernetwork φ_i. The hypernetwork generates the weight parameters by inputting corresponding vectors z_i. The client first aggregates hypernetwork information from neighbors to generate a new hypernet, then uses it to produce the shared part of model weights. It combines these with the previous round's personalized parameters to form the complete client model. Finally, the client performs local training to update φ_i, z_i and v_i and prepares for next round.

shared parameters, while training personalized parameters locally, which alleviates communication burden and maintains the ability to train diverse individual models.

We conduct experiments on non-IID settings across different data partitions (Dirichlet and Pathological distribution) and different partition coefficients. And then we compare the performance of our algorithm with many SOTA baselines in CIFAR10, CIFAR100, and Tiny-ImageNet datasets. Extensive evaluations of various classification tasks show that our algorithm can achieve competitive performance, with improvements in both communication cost and convergence performance. Our contributions are listed as follows.

- We propose a novel DPFL algorithm, DFedHP, which can effectively leverage the characteristics of hypernetworks to generate accurate personalized models. It reduces communication overhead significantly and fasters speed of convergence.
- We explore the impact of aggregating different numbers of node parameters on communication and convergence and demonstrate the compatibility and integrability of DFedHP with other PFL methods.
- We evaluate the communication and convergence performance with various data distributions. Extensive experiments prove that DFedHP achieves better performance effectively than SOTA PFL baselines.

2 Related Work

Decentralized Federated Learning (DFL). In DFL, there is no central server to coordinate and aggregate model updates, clients must communicate directly with each other. This enhances privacy but may lead to greater communication volume [Sun *et al.*, 2022; Shi *et al.*, 2023b; Li *et al.*, 2025b]. Besides, DFL primarily reduces communication pressure by optimizing communication content, reducing the frequency of communication and improving communication patterns [Liu *et al.*, 2022; Li *et al.*, 2025a], thus improving system efficiency and scalability. Because we mainly focus on the former, from this perspective, researchers typically use compression [Wang *et al.*, 2022] such as quantization, pruning [Xu *et al.*, 2024], or sparsification [Dai *et al.*, 2022] to compress the data that need to be transmitted and reduce the amount of data transmitted each time. For instance, ProxyFL[Kalra *et al.*, 2023] transmits a publicly shared proxy model to reduce communication costs and provide stronger privacy guarantees. KD-PDFL [Jeong and Kountouris, 2023] combines knowledge distillation with decentralized federated learning DFL, enhancing personalization capabilities and reducing transmission costs.

Personalized Federated Learning. PFL balances the generalization ability of the global model with the individualized needs of customers. Common methods include fine-tuning [Tamirisa *et al.*, 2024], knowledge distillation [Su *et al.*, 2025], regularization [Zhang *et al.*, 2023], parameter decoupling [Zhou *et al.*, 2024], and data augmentation etc. The parameter decoupling approach, such as FedPer [Arivazhagan *et al.*, 2019], FedRep [Collins *et al.*, 2021] and FedBABU [Oh *et al.*, 2021], generally allows each client retains a portion of the model parameters for local personalization while sharing other parameters for global aggregation. DisPFL [Dai

et al., 2022] and FedMask [Li *et al.*, 2021a] and achieve enhanced personalization and reduced communication costs by proposing a distributed sparse training technique. To achieve personalization, more complex models such as hierarchical [Ma *et al.*, 2022] and multi-task learning frameworks [Mills *et al.*, 2021] can be adopted, but this may increase computation and communication frequency. In addition, significant differences between PFL clients can require more frequent communication to ensure effective training participation [Pillutla *et al.*, 2022]. This motivates us to explore a PFL strategy to improve the communication burden.

Hypernetworks. Hypernetworks [Chauhan *et al.*, 2024] dynamically generate target network weights based on input data, rather than using pre-trained weights statically [Ha *et al.*, 2016]. It significantly reduces storage requirements without sacrificing too much performance and adapts to different input distributions or tasks. Hypernetworks are applied in few-shot learning [Sendera *et al.*, 2023], federated learning and reinforcement learning [Beck *et al.*, 2023]. In the field of federated learning, PFedHN [Shamsian *et al.*, 2021] deploys a hypernetwork on the central server using random fixed vectors to generate weights directly. PFedLA [Ma *et al.*, 2022] uses hypernetwork to generate weight matrices, and the server uses the matrix and client parameters to update personalized model weights. HFN [Chen *et al.*, 2024] deploys a hypernetwork on the client in CFL topologies, using a single vector from client's each layer and generate model weights. Different from current work, DFedHP ensures model performance while significantly reducing the communication frequency between participants, thereby improving system efficiency and response speed. It is particularly suitable for resource-constrained or applications that require rapid iteration and updates, such as the IoT and mobile edge computing.

3 Methodology

In this section, we first define the problem setting for decentralized partial personalized models. Next, we present the DFedHP algorithm, which aggregates the hypernetworks parameters of clients to reduce the communication burden and improve the convergence speed under resources heterogeneity situations while achieving SOTA performance.

3.1 Problem Setup

Considering n clients with parameters $\theta \in \mathbb{R}^d = \{\theta_1, ..., \theta_n\}$, let D_i be the non-IID data distribution of the client $i \in \{1, 2, ..., n\}$. The non-convex minimization problem in this work can be defined as:

$$\min_{\theta \in \mathbb{R}^d} F(\theta) := \frac{1}{n} \sum_{i=1}^{n} F_i(\theta_i), \tag{1}$$

where $F_i(\theta_i) = \mathbb{E}_{\xi \sim \mathcal{D}_i} F_i(\theta_i; \xi_i)$ refers to the local objective function of the client i, which is related to the data sample ξ_i of D_i. In order to preserve the personalized capabilities of the clients, we divide the model into two parts: the **shared parameters** and the **personal parameters** $v_i \in \mathbb{R}^{d_i}$. Different from other PFL algorithms, DFedHP replaces shared parameters with hypernetworks that have fewer parameters, which significantly saves the communication resources required for interaction among nodes. Specifically, $h(\varphi_i, z_i)$ denotes the the i_{th} client's hypernetwork with the hyper-parameters $\varphi_i \in \mathbb{R}^{d_z}$ and the embedding vectors $z_i \in \mathbb{R}^{d_z}$. Thus, the complete local model θ_i of the client i is denoted as $\theta_i = (h(\varphi_i, z_i), v_i)$. Consequently, the objective function can be derived from problem (1) to:

$$\min_{\varphi, z, V} F(h(\varphi, z), V) := \frac{1}{n} \sum_{i=1}^{n} F_i\left(h(\varphi, z), v_i\right), \tag{2}$$

where $V = \{v_1, v_2 ..., v_n\} \in \mathbb{R}^{d_1 + ... + d_n}$. φ presents the consensus hypernetwork model averaged with all hypernetworks φ_i from other clients, that is $\varphi = \frac{1}{n} \sum_{i=1}^{n} (\varphi_i)$. The communication network in the decentralized network topology among nodes is modeled as an undirected connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{V}, \mathbf{W})$. $\mathcal{N} = \{1, ..., n\}$ denotes the collection of nodes, $\mathcal{V} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the set of end-to-end communication channels. Consequently, the shared parameters of each client are sent to neighbors according to the mixing matrix $\mathbf{W}$, which defines the communication topology between clients. In contrast, personal parameters will not be sent out but will only be iterated locally.

3.2 DFedHP Algorithm

In this subsection, we introduce the DFedHP framework to solve problem (2) by deploying hypernetworks in a distributed network in a more efficient manner.

DFedPH Algorithm. The Figure 1 presents the main pipeline of DFedHP algorithm. Each client is composed of shared part and a personal part. The hypernetwork generates only the shared components of the local models. Rather than sharing their entire client models, clients only communicate hypernetwork parameters with their neighbors. After exchanging information, client aggregates the hyper-parameters of its neighbors and use it to generate shared part parameters. The clients combine local personalized parameters from the previous round of training to generate entire client model. Finally, client performs local training to update the complete parameters, and prepares to send the shared parameters for the next round. Specifically, in Algorithm 1, we form a complete local network $(h(\varphi_i, z_i), v_i)$ in line 6. In line 8 to line 19, multiple local iterative updates are performed on the personal parameters v_i, hypernetwork φ_i and vectors z_i in sequence. Finally, in line 19, DFedHP communicates with neighboring models according to mixing matrix $\mathbf{W}$ and aggregates hypernetworks. The client updates local hypernetwork by computing the average of hyper-parameters from neighbors (including itself) for the next round.
Consequently, communication between clients can mainly transmit these smaller hypernetwork parameters, rather than the complete structure or model weights. This greatly reduces the amount of communication data required for each iteration. In addition, the hypernetwork serves as an intermediate layer, providing an additional layer of abstraction for sensitive information, further enhancing privacy. Moreover, based on the data characteristics, hypernetwork generates model initialization weights that are closer to the specific task requirements, thereby achieving faster convergence. Furthermore,

in addition to DFedAvg, DFedHP can seamlessly integrate into existing DPFL training algorithms, with its lightweight framework and flexible modular design.

Hypernetwork Structure. The hypernetwork maps a certain form of a small input to the weights of the target network. The input generates specific model weights for different tasks, clients, or data set characteristics, including random noise, fixed or dynamically varying vectors [He *et al.*, 2023; Littwin *et al.*, 2020]. In this paper, we input learnable embbeding vectors z_i into the hypernetwork φ, we can obtain output $h(\varphi_i, z_i)$ of the client i. If we input a vector into the hypernet to generate the parameters, we would encounter the problem of excessive hyper-parameters and high communication costs [Ha *et al.*, 2016]. Therefore, we input a large number of different vectors to generate multiple filters, and by concatenating different outputs, we form a large-sized weight matrix. This can significantly reduce the number of hypernetwork parameters, which is far lower than the parameter numbers of original network. As the hypermodel parameters are mainly transmitted between clients, our method greatly reduces the communication cost in DFL. In this paper, we mainly discuss the case of hypernetwork composed of two linear layer and activation function. In this configuration, DFedHP achieves a reduction of up to 87.81% in the number of parameters transmitted per round. Smaller hypernet parameters are transmitted between clients, rather than the full model weights, which reduces the communication costs and time.

Partial Personalized Models [Liu *et al.*, 2025]**.** In a neural network, the convolutional layers near the input extract features representations of input by filters (also called kernels or feature detectors), mapping data to easily discernible low-dimensional spaces [Sabah *et al.*, 2024; Tan *et al.*, 2022]. For many tasks, such as image classification, the early convolutional layers often capture some low-level features, such as edges and textures, which have high generality. Therefore, by sharing the convolutional layers, effective low-level feature extractors can be shared among all clients, thereby enhancing resource utilization efficiency and model performance. This shared part is generated by the hypernetwork in this paper. Conversely, the linear layer close to the output primarily maps features to a specific output space. It is focused on pattern recognition to determine the data category, which we designate as the personalized part. The linear layers can be personalized according to the data characteristics of each client to better adapt to the local data distribution.

4 Theoretical Analysis

In this section, we conduct a convergence analysis of the DFedHP algorithm and explore its working principle. First, we make some general assumptions.

Definition 1 (The gossip/mixing matrix [Sun *et al.*, 2022]). The gossip matrix $\mathbf{W} = [w_{i,j}] \in [0,1]^{n\times n}$ is assumed to have these properties: (i) (Graph) If $i \neq j$ and $(i,j) \notin \mathcal{V}$ then $w_{i,j} = 0$, otherwise, $w_{i,j} > 0$; (ii) (Symmetry) $\mathbf{W} = \mathbf{W}^\top$; (iii) (Null space property) null $\{\mathbf{I} - \mathbf{W}\}$ =span$\{\mathbf{1}\}$; (iv) (Spectral property) $\mathbf{I} \succeq \mathbf{W} \succ -\mathbf{I}$. With these properties, the eigenvalues of W satisfy $1 = \lambda_1(\mathbf{W})) > \lambda_2(\mathbf{W})) \geq \cdots \geq \lambda_n(\mathbf{W})) > -1$. In addition, $\lambda := \max\{|\lambda_2(\mathbf{W})|, |\lambda_n(\mathbf{W}))|\}$ and $1-\lambda \in (0,1]$ denote the spectral gap of $\mathbf{W}$, which quantifies the characteristics of the network topology.

Algorithm 1 DFedHP

Input: total number of devices n and communication rounds T. Learning rate: personal part η_v, hypernet η_φ and embedding vectors η_z. Number of local iterates K_v, K_φ and K_z.
Output: hyper parameters φ_i^T, personal part v_i^T and embedding vectors z_i^T after the final communication of all clients.
Initialization: randomly initialize each device's hypernet parameters φ_i^0, personal parameters v_i^0 and vectors z_i^0.
for each communication round $t \to 1$ to T **do**
  for client i **in parallel** do **do**
    Generate model parameters by φ_i^0, sample a batch of local data ξ_i and calculate local gradient iteration.
    for local epoch from k = 0 to K_v do **do**
      Perform personal parameters v_i update:
      $v_i^{t,k+1} \leftarrow v_i^{t,k} - \eta_v \nabla_v F_i \left(h(\varphi_i^t, z_i^t), v_i^{t,k}\right)$
      $v_i^{t+1} \leftarrow v_i^{t,K_v}$
    end for
    for local epoch from k=0 to K_φ do **do**
      Update the hypernetworks parameters φ_i:
      $\varphi_i^{t,k+1} \leftarrow \varphi_i^{t,k} - \eta_\varphi \nabla_\varphi F_i \left(h(\varphi_i^{t,k}, z_i^t), v_i^{t+1}\right)$
    end for
    $g_{1,i}^t \leftarrow \varphi_i^{t,K_\varphi}$
    for local epoch from k=0 to K_φ do **do**
      Update the embedded vector z_i:
      $z_i^{t,k+1} \leftarrow z_i^{t,k} - \eta_z \nabla_z F_i \left(h(\varphi_i^{t,k+1}, z_i^{t,k}), v_i^{t+1}\right)$
    end for
    $g_{2,i}^t \leftarrow z_i^{t,K_z}$
    Receive neighbors' hyper-network $g_{1,j}^t$, $g_{2,j}^t$.
    with mixing matrix $\mathbf{W}$:
    $\varphi_i^{t+1} = \sum_{l\in\mathcal{N}(i)} w_{i,l} g_{1,i}^t$, $z_i^{t+1} = \sum_{l\in\mathcal{N}(i)} w_{i,l} g_{2,i}^t$
  end for
end for

Assumption 1 (Lipschitz Smoothness). F_i and h_i are continuously differentiable. There exist constants L_φ, L_z, L_h, L_v,L_{hv}, L_{vh}, $L_{h\varphi}$, L_{hz}, $\forall i \in \{1,2,...,n\}$ such that:

- $\nabla_h F_i(h_i, v_i)$ is $L_h - Lipschitz$ with respect to h_i, $L_{hv} - Lipschitz$ with respect to v_i;
- $\nabla_v F_i(h_i, v_i)$ is $L_v - Lipschitz$ with respect to v_i, $L_{vh} - Lipschitz$ with respect to h_i;
- $\nabla_\varphi F_i(h(\varphi_i, z_i), v_i)$ is $L_\varphi - Lipschitz$ with respect to φ_i and $\nabla_z F_i(h(\varphi_i, z_i), v_i)$ is $L_z - Lipschitz$ with respect to z_i;
- $\nabla_\varphi h_i$ is $L_{h\varphi} - Lipschitz$ continuous with respect to φ_i and $\nabla_z h_i$ is $L_{hz} - Lipschitz$ continuous with respect to z_i.

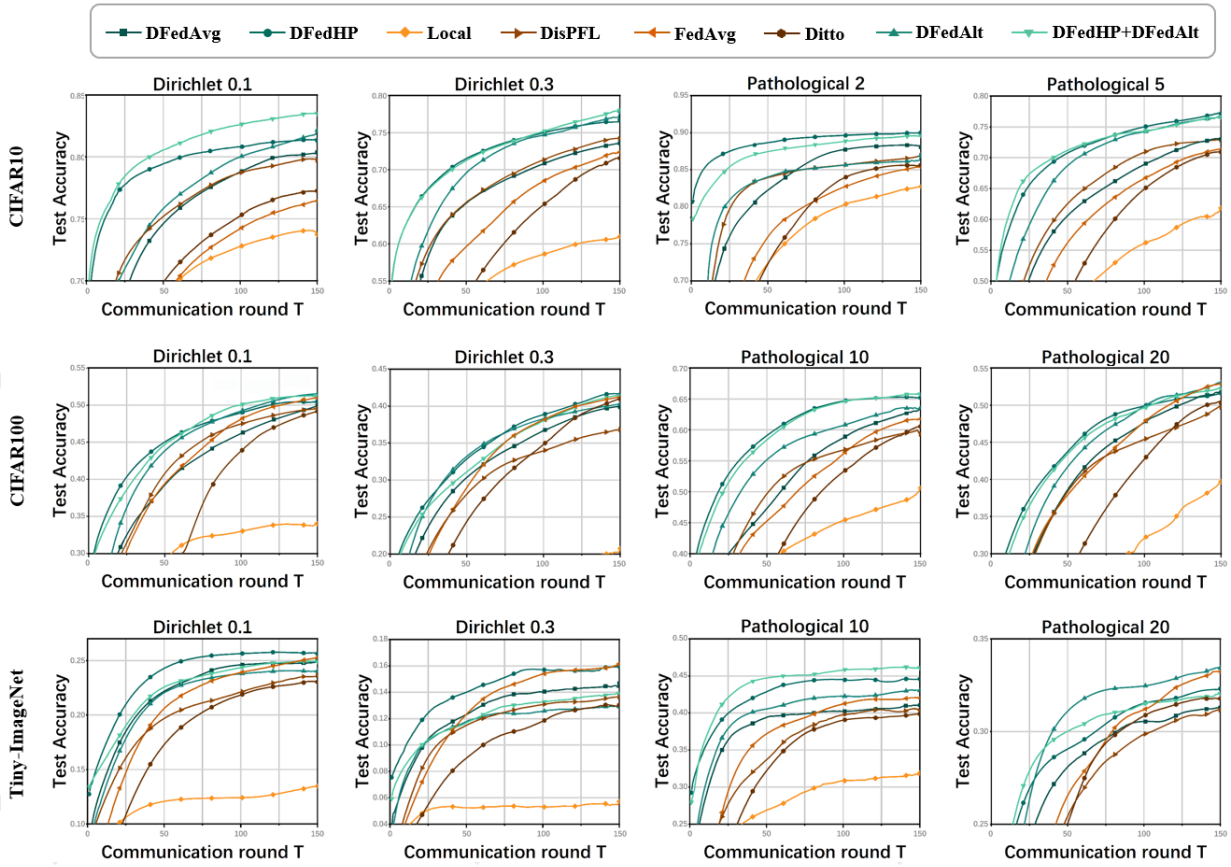

Figure 2: Test accuracy on CIFAR-10, CIFAR-100, Tiny-ImageNet datasets with heterogenous data partitions.

The gradients $\nabla_\varphi h$ and $\nabla_z h$ are respectively bounded by constants γ_φ and γ_z. Besides, we present the relative cross-sensitivity of gradient with the scalar:

$$\chi := \max\{L_{hv}, L_{vh}\} / \sqrt{L_h L_v}.. \tag{3}$$

Assumption 2 (Bounded Variance). The gradient of the function F_i has bounded variance. There exist constants σ_φ, σ_z and σ_v , $\forall i \in \{1, 2, ..., n\}$ such that:

- $\mathbb{E}\left[\left\|\nabla_\varphi F_i\left(h_i, v_i; \xi_i\right) - \nabla_\varphi F_i\left(h_i, v_i\right)\right\|^2\right] \leq \sigma_\varphi^2$;
- $\mathbb{E}\left[\left\|\nabla_z F_i\left(h_i, v_i; \xi_i\right) - \nabla_z F_i\left(h_i, v_i\right)\right\|^2\right] \leq \sigma_z^2$;
- $\mathbb{E}\left[\left\|\nabla_v F_i\left(h_i, v_i; \xi_i\right) - \nabla_v F_i\left(h_i, v_i\right)\right\|^2\right] \leq \sigma_v^2$.

Assumption 3 (Partial Gradient Diversity). There exists a constant $\delta \geq 0, \forall \varphi_i, z_i, V$ such that:

- $\frac{1}{n}\sum_{i=1}^n \left\|\nabla_\varphi F_i\left(h_i, v_i\right) - \nabla_\varphi F\left(h_i, V\right)\right\|^2 \leq \delta_\varphi^2$;
- $\frac{1}{n}\sum_{i=1}^n \left\|\nabla_z F_i\left(h_i, v_i\right) - \nabla_z F\left(h_i, V\right)\right\|^2 \leq \delta_z^2$.

Theorem 1 (Convergence Analysis for DFedHP). *Assume Assumptions 1-3 holds, let the local adaptive learning rate satisfy* $\eta_\varphi = O(1/L_\varphi K_\varphi \sqrt{T})$, $\eta_z = O(1/L_z K_z \sqrt{T})$, $\eta_v = O(1/L_v K_v \sqrt{T})$, *where T is the number of communication rounds. Furthermore, we adopt the averaged parameter* $\bar{\varphi}^t = \frac{1}{n}\sum_{i=1}^n (\varphi_i^t)$, $\bar{z}^t = \frac{1}{n}\sum_{i=1}^n (z_i^t)$ *of all clients as an approximate solution to the problem* (2) *below. Additionally,* F^* *denotes the minimal value of* F: $F(h(\bar{\varphi}, \bar{z}), V) \geq F^*$ *for all* φ, z, *and* $\Delta_{\bar{\varphi}}^t$, $\Delta_{\bar{z}}^t$ *and* $\Delta_{\bar{v}}^t$ *are performed as:*

$$\begin{aligned}
\Delta_{\bar{\varphi}}^t &= \left\|\nabla_\varphi F\left(h\left(\bar{\varphi}^t, \bar{z}^t\right), V^{t+1}\right)\right\|^2, \\
\Delta_{\bar{z}}^t &= \left\|\nabla_z F\left(h\left(\bar{\varphi}_i^t, \bar{z}_i^t\right), V^{t+1}\right)\right\|^2, \\
\Delta_v^t &= \frac{1}{n}\sum_{i=1}^n \left\|\nabla_v F_i\left(h\left(\varphi_i^t, z_i^t\right), v_i^t\right)\right\|^2.
\end{aligned} \tag{4}$$

Thus, we present the convergence rate:

$$\begin{aligned}
&\frac{1}{T}\sum_{i=1}^T \left(\frac{1}{L_\varphi}\mathbb{E}\left[\Delta_{\bar{\varphi}}^t\right] + \frac{1}{L_z}\mathbb{E}\left[\Delta_{\bar{z}}^t\right] + \frac{1}{L_v}\mathbb{E}\left[\Delta_{\bar{v}}^t\right]\right) \\
&\leq \mathcal{O}\left(\frac{F\left(h(\bar{\varphi}^1, \bar{z}^1), V^1\right) - F^*}{\sqrt{T}} + \frac{\sigma_1^2}{(1-\lambda)^2\sqrt{T}} + \frac{\sigma_2^2}{\sqrt{T}} + \frac{\sigma_3^2}{(1-\lambda)^2 T}\right),
\end{aligned} \tag{5}$$

where the constants σ_1^2, σ_2^2 *and* σ_3^2 *are defined as:*

$$\begin{aligned}
\sigma_1^2 &= \frac{\chi^2 L_v L_\varphi + 2K_\varphi \gamma_\varphi^2}{L_\varphi^2}\left(\sigma_\varphi^2 + \delta_\varphi^2\right) \\
&\quad + \frac{\chi^2 L_v L_z + 2K_z \gamma_z^2}{L_z^2}\left(\sigma_z^2 + \delta_z^2\right), \\
\sigma_2^2 &= \frac{\sigma_v^2(L_v+1)}{L_v^2} + \frac{(\sigma_\varphi^2+\delta_\varphi^2)L_{h\varphi}}{K_\varphi L_\varphi} + \frac{(\sigma_z^2+\delta_z^2)L_{hz}}{K_z L_z}, \\
\sigma_3^2 &= \frac{(\sigma_\varphi^2+\delta_\varphi^2)\gamma_\varphi^2 L_{h\varphi}^2}{K_\varphi L_\varphi} + \frac{(\sigma_z^2+\delta_z^2)\gamma_z^2 L_{hz}^2}{K_z L_z}.
\end{aligned} \tag{6}$$

Remark 1 (Statistical Heterogeneity Affects the Convergence Boundary). *The smaller local variance variables* $\sigma_\varphi^2, \sigma_z^2$, *smaller gradient diversity* δ_φ,δ_z *and smoother Lipschitz constants all lead to a tighter convergence boundary.*

Algorithm	CIFAR-10				CIFAR-100				Tiny-ImageNet			
	Dirichlet		Pathological		Dirichlet		Pathological		Dirichlet		Pathological	
	$\alpha = 0.1$	$\alpha = 0.3$	$c = 2$	$c = 5$	$\alpha = 0.1$	$\alpha = 0.3$	$c = 10$	$c = 20$	$\alpha = 0.1$	$\alpha = 0.3$	$c = 10$	$c = 20$
Local	74.83	61.54	83.77	63.42	34.93	21.89	50.64	40.68	13.83	5.72	31.90	18.24
FedAvg	76.34	72.58	85.12	72.61	51.02	41.05	61.98	53.39	25.26	16.17	42.93	33.51
Ditto	77.21	71.95	85.36	72.04	49.43	40.95	61.30	51.41	24.08	13.31	39.85	32.08
FedPer	82.47	77.20	89.01	77.35	48.98	38.36	62.15	52.84	23.53	10.08	45.28	32.62
DisPFL	79.93	74.12	86.99	73.78	49.77	37.28	60.33	50.23	24.16	13.46	41.02	32.04
DFedAvg	80.04	73.96	87.83	74.35	50.65	40.61	63.59	52.92	25.07	14.90	41.74	31.97
DFedAlt	82.65	77.91	86.42	76.60	51.81	40.70	63.02	53.68	24.30	13.19	44.12	33.50
DFedHP	81.79	77.01	89.64	77.43	50.08	41.54	65.06	52.70	26.22	15.93	44.06	32.16
DisPFL+HP	80.48	75.18	89.55	74.31	49.26	40.38	61.52	50.86	23.29	13.05	42.23	32.50
DFedAlt+HP	83.51	78.04	89.23	76.53	51.16	41.24	65.18	52.97	24.92	13.86	46.55	32.34

Table 1: Test accuracy on CIFAR-10, CIFAR-100 and Tiny-ImageNet in both Dirichlet (α) and Pathological (c) data distribution settings.

Since in a round of communication, $v_i$is updated first, followed by φ_i and then z_i, there is no L_{vh} in the convergence boundary. The proof details are in the ***Appendix***.

5 Experiments

In this section, we introduce the experiment, covering the setup and the performance of the DFedHP. Additionally, we present the communication advantages of the method and verify its effectiveness when combined with other methods.

5.1 Experiment Setup

Dataset and Data Partition. We evaluate the performance of DFedHP on the CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets under Dirichlet distribution and Pathological distribution. The Dirichlet distribution specifies that the local data set conforms to the Dirichlet distribution. And the Pathological distribution is to allocate a finite number of classifications to each client. The two types of approaches guarantee the distribution of all data sets among the clients is non-IID. We partition the training and testing data according to the Dirichlet distribution Dir(α). The smaller partition alpha α is, the more uneven the data distribution among clients will be, resulting in higher data heterogeneity [Kotelevskii *et al.*, 2023; Wang *et al.*, 2020]. In addition, we sample different classes from the dataset for each client. The fewer classes each client has, the more heterogeneous the setting becomes.

Baselines and Backbone. We evaluate our method and the following SOTA baselines of DFL. We select FedAvg [McMahan *et al.*, 2017], FedPer [Arivazhagan *et al.*, 2019], and Ditto [Li *et al.*, 2021b] as the FL baseline, and DFedAvg [Sun *et al.*, 2022] and DFedAlt [Shi *et al.*, 2023a] as the DFL baseline. Local means only local training for each local client with no communication. Note that we define the accuracy as the personal test accuracy. The PFL methods used in the experiment learns personal classifiers for each client based on a shared feature extraction.

Training Strategies. In all experiments, all algorithms are conducted on ResNet-18 [He *et al.*, 2016] with batch normalization. We record the communication between the client and server (or between clients) at 150 rounds. The total number of clients is 100 and the communication radio is 0.1 for each round. The batch size is 128 and the local training epoch is 4. The experiment uses SGD as the optimizer with momentum of 0.9. The distributed topologies is random. For DFedHP, the embedding vector size is 128. Although dimensions of kernels in each layer are different, the dimension of a kernel is often an integer multiple of a fixed value [Ha *et al.*, 2016]. We choose 64 as the fixed value of ResNet-18. We conduct multiple experiments on the learning rates of all algorithms.

Algorithm	CIFAR-100		Tiny-ImageNet	
	Dirichlet	Pathological	Dirichlet	Pathological
	$\alpha = 0.3$	c =20	$\alpha = 0.3$	$c = 20$
	@85	@85	@80	@90
FedAvg	74	89	70	79
DisPFL	73	76	45	96
DFedAvg	79	84	47	73
DFedHP	62	51	27	63

Table 2: The communication rounds necessary for attaining the target accuracy.

5.2 Performance Evaluation

Figure 2 shows the convergence curves of DFedHP and baseline methods in image classification tasks using the ResNet-18 model on different datasets and different non-IID data distributions. DFedHP significantly reduces communication costs. In the experimental setup of the paper, the amount of data each client needs to transmit is reduced from a maximum 43,666.97KB to 5,321.06KB (32 float) in Table 3. Under the decentralized setting, we assume that each client communicates with 10 neighbors in each round, following the hypernetwork configuration in the main experiment. Each communication transmits at least 50.75 GB of parameter information transmitted in 500 rounds of communication. Without the hypernetwork, the minimum communication volume in 500 rounds is at least 418.23 GB. In sum, due to the use of hypernetwork weight generation network, the client only needs to transmit a smaller number of hypernetwork parameters, thus reducing the communication burden.

Costs	com.(KB)	computation (s)	memory (MB)
DfedHP	5.32×10^3	2.51×10^4	604.68 / 21.63
DFedAvg	4.37×10^4	2.39×10^4	669.87 / 103.06

Table 3: A comparison of each client's communication, computation, and (maximum and average) video memory costs per round.

In addition, Table 2 presents the number of communication rounds required to achieve the DFedHP target accuracy. It is evident that DFedHP shows significant improvements in convergence speed. In Pathological 20, the convergence speed of DFedHP in CIFAR-100 is 39.29 % faster than that of DFedAvg, and the convergence speed of the meta-network transmission is greatly improved. Moreover, DFedHP can be used with any PFL method, such as "DisPFL + HP", "DFedAlt + HP" in Table 1. In most cases, applying DFedHP to other DFL frameworks significantly reduces convergence speed and communication overhead. Faster convergence enables the model to reach optimal or satisfactory performance more quickly, reducing required iterations. This is crucial in poor network conditions, where fewer communication rounds lower overall costs. In applications that need rapid adaptation to new data or changing environments, faster convergence allows timely model updates and more accurate services.

Then, we compare the test accuracy of all baselines under different settings in Table 1. In CIFAR-10 dataset, DFedHP achieves 77.01% in the Dirichlet 0.3 setting, which is 3.05% higher than the method without hypernetwork (DFedavg). DFedHP performs well on the Dirichlet distribution and pathological distribution. It can resist the negative effects of some data heterogeneity, which proves the effectiveness and robustness of the proposed method. Moreover, in cases of excessive data heterogeneity, due to the limited local training data, accuracy decreases as the level of heterogeneity decreases, which is a normal phenomenon. It performs well in scenarios with data heterogeneity, which proves the effectiveness of the proposed method. DFedHP utilizes embedded vectors to capture more precise features of the data. This allows the classification head to be more adaptive to the local data of each client through a stronger feature extractor, thereby enabling the meta-network to generate accurate model weights. In summary, the effects of DFedHP are significant, and it can reduce communication burden and achieve faster convergence speed with greater benefits in distributed scenarios when performing partial model personalization.

5.3 Ablation Study

We conduct several ablation study on CIFAR-10 with Dirichlet 0.3 distribution to verify the validity and robustness of the component. In Figure 3 (a), FedHP achieves the robustness in various communication topologies. Furthermore, based on the spectral gap analysis in **Section** 4, as the sparsity of topology decreases, spectral gap decreases, the generalization effect improves. The results in Figure 3(a) are consistent with the analysis. The communication costs and convergence boundaries of different topologies, from largest to smallest, are: fully connected, exponential, grid, ring. Figure 3(b) shows the impact of the number of hypernetwork

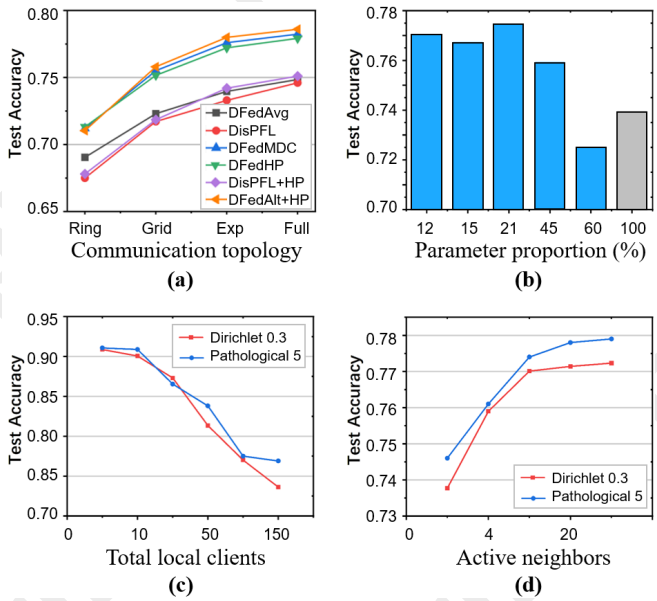

Figure 3: Ablation study on CIFAR-10 with Dirichlet 0.3 distribution. (a) Personal test accuracy in various network topologies for DFL algorithms ; (b) the numbers of hypernetwork parameters; (c) the number of total clients; (d) the number of active neighbors per round of communication;

parameters on model accuracy under different network structures. The X-axis represents the proportion of hypernetwork parameters in the original model without hypernetwork. In the experiment, the best accuracy results among different hypernetwork structures with the same number of hyperparameters are selected. The number of parameters in a HyperNetwork is not fixed, as it depends on model complexity, task requirements, dataset characteristics, and expected performance. It must have enough expressive power to generate effective weights while avoiding overfitting or wasting computational resources. In addition, more detailed ablation studies are shown in the **Appendix**.

6 Conclusion

In this work, we propose a new DPFL framework, DFedHP, to realize decentralized personalized model aggregation through hyper network aggregation. This enhances collaboration among clients and generates better personalized models on non-IID datasets. In addition, DFedHP decouples training complexity from communication complexity. The size of the generated client models is not limited, effectively reducing communication overhead. Further, DFedHP can be combined with other related methods, which is scalable and flexible. It can adapt to clients with different computing capabilities or large-scale federated learning scenarios, where communication capacity is typically limited. Extensive evaluations on various classification tasks have demonstrated that the communication efficiency and convergence ability of DFedHP significantly enhance. In the future, we will explore the theoretical analysis and application extensively.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62403484.

Contribution Statement

Corresponding authors: Yong Peng and Xiaoyan Wang.

References

[Arivazhagan *et al.*, 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers, 2019.

[Beck *et al.*, 2023] Jacob Beck, Matthew Thomas Jackson, Risto Vuorio, and Shimon Whiteson. Hypernetworks in meta-reinforcement learning. In *Conference on Robot Learning*, pages 1478–1487. PMLR, 2023.

[Beltrán *et al.*, 2023] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 2023.

[Caldarola *et al.*, 2022] Debora Caldarola, Barbara Caputo, and Marco Ciccone. Improving generalization in federated learning by seeking flat minima. In *European Conference on Computer Vision*, pages 654–672. Springer, 2022.

[Chauhan *et al.*, 2024] Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. A brief review of hypernetworks in deep learning. *Artificial Intelligence Review*, 57(9):250, 2024.

[Chen *et al.*, 2024] Xingyun Chen, Yan Huang, Zhenzhen Xie, and Junjie Pang. Hyperfednet: Communication-efficient personalized federated learning via hypernetwork. *arXiv preprint arXiv:2402.18445*, 2024.

[Collins *et al.*, 2021] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

[Dai *et al.*, 2022] Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. Dispfl: Towards communication-efficient personalized federated learning via decentralized sparse training. In *International conference on machine learning*, pages 4587–4604. PMLR, 2022.

[Ha *et al.*, 2016] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[He *et al.*, 2023] Yunlong He, Dandan Yan, and Fei Chen. Hierarchical federated learning with local model embedding. *Engineering Applications of Artificial Intelligence*, 123:106148, 2023.

[Jeong and Kountouris, 2023] Eunjeong Jeong and Marios Kountouris. Personalized decentralized federated learning with knowledge distillation. In *ICC 2023-IEEE International Conference on Communications*, pages 1982–1987. IEEE, 2023.

[Kalra *et al.*, 2023] Shivam Kalra, Junfeng Wen, Jesse C Cresswell, Maksims Volkovs, and Hamid R Tizhoosh. Decentralized federated learning through proxy model sharing. *Nature communications*, 14(1):2899, 2023.

[Kotelevskii *et al.*, 2023] Nikita Kotelevskii, Samuel Horváth, Karthik Nandakumar, Martin Takáč, and Maxim Panov. Dirichlet-based uncertainty quantification for personalized federated learning with improved posterior networks. *arXiv preprint arXiv:2312.11230*, 2023.

[Li *et al.*, 2021a] Ang Li, Jingwei Sun, Xiao Zeng, Mi Zhang, Hai Li, and Yiran Chen. Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 42–55, 2021.

[Li *et al.*, 2021b] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pages 6357–6368. PMLR, 2021.

[Li *et al.*, 2023] Bo Li, Mikkel N Schmidt, Tommy S Alstrøm, and Sebastian U Stich. On the effectiveness of partial variance reduction in federated learning with heterogeneous data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2023.

[Li *et al.*, 2025a] Qinglun Li, Li Shen, Guanghao Li, Quanjun Yin, and Dacheng Tao. Dfedadmm: Dual constraint controlled model inconsistency for decentralize federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[Li *et al.*, 2025b] Qinglun Li, Miao Zhang, Tao Sun, Quanjun Yin, and Li Shen. Dfedgfm: Pursuing global consistency for decentralized federated learning via global flatness and global momentum. *Neural Networks*, 184:107084, 2025.

[Littwin *et al.*, 2020] Etai Littwin, Tomer Galanti, Lior Wolf, and Greg Yang. On infinite-width hypernetworks. *Advances in neural information processing systems*, 33:13226–13237, 2020.

[Liu *et al.*, 2022] Wei Liu, Li Chen, and Wenyi Zhang. Decentralized federated learning: Balancing communication and computing costs. *IEEE Transactions on Signal and Information Processing over Networks*, 8:131–143, 2022.

[Liu *et al.*, 2025] Yingqi Liu, Qinglun Li, Jie Tan, Yifan Shi, Li Shen, and Xiaochun Cao. Understanding the stability-based generalization of personalized federated learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Ma *et al.*, 2022] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10092–10101, June 2022.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Mills *et al.*, 2021] Jed Mills, Jia Hu, and Geyong Min. Multi-task federated learning for personalised deep neural networks in edge computing. *IEEE Transactions on Parallel and Distributed Systems*, 33(3):630–641, 2021.

[Oh *et al.*, 2021] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021.

[Pillutla *et al.*, 2022] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *International Conference on Machine Learning*, pages 17716–17758. PMLR, 2022.

[Sabah *et al.*, 2024] Fahad Sabah, Yuwen Chen, Zhen Yang, Muhammad Azam, Nadeem Ahmad, and Raheem Sarwar. Model optimization techniques in personalized federated learning: A survey. *Expert Systems with Applications*, 243:122874, 2024.

[Sendera *et al.*, 2023] Marcin Sendera, Marcin Przewiez likowski, Konrad Karanowski, Maciej Zieba, Jacek Tabor, and Przemyslaw Spurek. Hypershot: Few-shot learning by kernel hypernetworks. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2469–2478, 2023.

[Shamsian *et al.*, 2021] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021.

[Shi *et al.*, 2023a] Yifan Shi, Yingqi Liu, Yan Sun, Zihao Lin, Li Shen, Xueqian Wang, and Dacheng Tao. Towards more suitable personalization in federated learning via decentralized partial model training, 2023.

[Shi *et al.*, 2023b] Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. Improving the model consistency of decentralized federated learning. In *International Conference on Machine Learning*, pages 31269–31291. PMLR, 2023.

[Su *et al.*, 2025] Liwei Su, Donghao Wang, and Jinghua Zhu. Dkd-pfed: A novel framework for personalized federated learning via decoupling knowledge distillation and feature decorrelation. *Expert Systems with Applications*, 259:125336, 2025.

[Sun *et al.*, 2022] Tao Sun, Dongsheng Li, and Bao Wang. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4289–4301, 2022.

[Tamirisa *et al.*, 2024] Rishub Tamirisa, Chulin Xie, Wenxuan Bao, Andy Zhou, Ron Arel, and Aviv Shamsian. Fedselect: Personalized federated learning with customized selection of parameters for fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23985–23994, 2024.

[Tan *et al.*, 2022] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603, 2022.

[Wang *et al.*, 2020] Yansheng Wang, Yongxin Tong, and Dingyuan Shi. Federated latent dirichlet allocation: A local differential privacy based framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6283–6290, 2020.

[Wang *et al.*, 2022] Lun Wang, Yang Xu, Hongli Xu, Min Chen, and Liusheng Huang. Accelerating decentralized federated learning in heterogeneous edge computing. *IEEE Transactions on Mobile Computing*, 22(9):5001–5016, 2022.

[Wu *et al.*, 2022] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Communication-efficient federated learning via knowledge distillation. *Nature communications*, 13(1):2032, 2022.

[Xu *et al.*, 2024] Yin Xu, Mingjun Xiao, Jie Wu, Guoju Gao, Datian Li, Haotian Xu, and Tongxiao Zhang. Enhancing decentralized federated learning with model pruning and adaptive communication. *IEEE Transactions on Industrial Informatics*, 2024.

[Yuan *et al.*, 2024] Liangqi Yuan, Ziran Wang, Lichao Sun, S Yu Philip, and Christopher G Brinton. Decentralized federated learning: A survey and perspective. *IEEE Internet of Things Journal*, 2024.

[Zhang *et al.*, 2023] Hao Zhang, Chenglin Li, Wenrui Dai, Junni Zou, and Hongkai Xiong. Fedcr: Personalized federated learning based on across-client common representation with conditional mutual information regularization. In *International Conference on Machine Learning*, pages 41314–41330. PMLR, 2023.

[Zhou *et al.*, 2024] Xu Zhou, Jie Li, Gongjin Lan, Rongrong Ni, Angelo Cangelosi, Jiaxin Wang, and Xiaofeng Liu. Efficient lower layers parameter decoupling personalized federated learning method of facial expression recognition for home care robots. *Information Fusion*, 106:102261, 2024.

[Zhu *et al.*, 2024] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024.