

Decentralized Directed Collaboration for Personalized Federated Learning

Yingqi Liu¹ Yifan Shi² Qinglun Li³ Baoyuan Wu⁴ Xueqian Wang² Li Shen^{5*}

¹Nanjing University of Science and Technology, Nanjing, China; ²Tsinghua University, Shenzhen, China;

³National University of Defense Technology, Changsha, China;

⁴The Chinese University of Hong Kong, Shenzhen, China; ⁵JD Explore Academy; Beijing, China.

lyq@njust.edu.cn; shiyf21@mails.tsinghua.edu.cn; liqinglun@nudt.edu.cn;

wubaoyuan@cuhk.edu.cn; wang.xq@sz.tsinghua.edu.cn; mathshenli@gmail.com.

Abstract

Personalized Federated Learning (PFL) is proposed to find the greatest personalized models for each client. To avoid the central failure and communication bottleneck in the server-based FL, we concentrate on the Decentralized Personalized Federated Learning (DPFL) that performs distributed model training in a Peer-to-Peer (P2P) manner. Most personalized works in DPFL are based on undirected and symmetric topologies, however, the data, computation and communication resources heterogeneity result in large variances in the personalized models, which lead the undirected aggregation to suboptimal personalized performance and unguaranteed convergence. To address these issues, we propose a directed collaboration DPFL framework by incorporating stochastic gradient push and partial model personalized, called **Decentralized Federated Partial Gradient Push (DFedPGP)**. It personalizes the linear classifier in the modern deep model to customize the local solution and learns a consensus representation in a fully decentralized manner. Clients only share gradients with a subset of neighbors based on the directed and asymmetric topologies, which guarantees flexible choices for resource efficiency and better convergence. Theoretically, we show that the proposed DFedPGP achieves a superior convergence rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$ in the general non-convex setting, and prove the tighter connectivity among clients will speed up the convergence. The proposed method achieves state-of-the-art (SOTA) accuracy in both data and computation heterogeneity scenarios, demonstrating the efficiency of the directed collaboration and partial gradient push.

1. Introduction

Recently, Personalized Federated Learning (PFL) has emerged to find the best model for each client since one consensus model can not satisfy all clients' needs in classi-

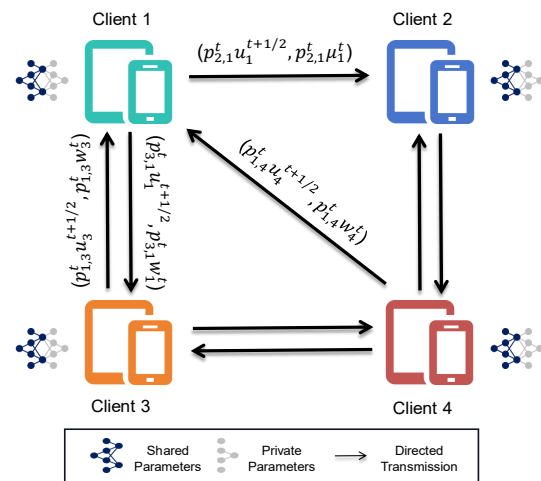


Figure 1. An overview of the DFedPGP with a directed graph. We take Client 1 as an example. It pushes the shared parameters $p_{j,1}^t, u_1^{t+1/2}$ and bias information $p_{j,1}^t, \mu_1^t$ to its out-neighbors (Client 2, 3); pulls the shared parameters $p_{1,j}^t, u_j^{t+1/2}$ and bias information $p_{1,j}^t, \mu_j^t$ from its in-neighbors (Client 3, 4).

cal Federated Learning (FL) [69]. The existing PFL algorithm can be categorized into two branches in terms of the existence of the centralized server (i.e., Centralized Personalized Federated Learning (CPFL) [1, 22, 39, 46] and Decentralized Personalized Federated Learning (DPFL) [14, 31, 55]). The challenges of centralized communication bottleneck or central failure may incur low communication efficiency or system crash in the federated processing. Thus, we focus on the DPFL, which allows edge clients to communicate with each other in a peer-to-peer manner, aiming to reduce the communication column of the busiest server node and embrace peer-to-peer communication for faster convergence. In decentralized FL, clients usually follow an undirected and symmetric communication topology to reach a consensus model [14, 53, 56], which means if one client receives neighbors' models, it sends its model back.

*Corresponding Author.

In order to satisfy the unique needs of individual clients, most existing works in PFL carefully designed the relationships between the global model and personalized models to fit the local data distribution via different techniques, such as parameter decoupling [1, 13, 46], knowledge distillation [19, 32, 39], multi-task learning [22, 54], model interpolation [15, 16] and clustering [17, 50]. These techniques can also be adopted to improve the personalized performance in DPFL [14, 36]. However, the heterogeneity among clients exists not only in local data distribution but also in the communication power and computation resources [7–9]. The power level of the wireless channel among clients may be different and time-varying in communication networks, and some clients may get offline occasionally without sending messages to their neighbors. These result in long-term waits or incidents of deadlock for their neighbors [11, 70] and also lead to poor convergence for the whole system. Besides, there is no reason to expect that the exchanged models are trained at the same convergence level due to the heterogeneous computation resources. Clients may receive excessive poor-performing models which can not help their training and degrade the personalized performance.

To tackle the challenges above, we propose a DPFL framework with a directed communication topology, termed DFedPGP, which incorporates the partial model personalization and stochastic gradient push to boost the personalized performance of the heterogeneous clients. Both partial model personalization and stochastic gradient push contribute to speeding up the convergence and reducing the communication resources to reach an ideal performance. Instead of exchanging the full model with their undirected neighbors, we decouple the model as a mixture of a shared feature representation part and a private linear classifier part and only push the shared partial gradients to the directed out-neighbors (as depicted in Figure 1). Specifically, the proposed method consists of three steps: (1) pull the shared partial gradient and the bias weights from in-neighbors; (2) local update the personalized linear classifier and the shared feature representation alternately with the de-biased parameters; (3) push the updated shared gradients and the bias information to out-neighbors. In-neighbors and out-neighbors are the in-coming and out-coming links for each client here. Partial gradient push makes the personalized information well stored in the private linear classifier, reducing communication costs as well as protecting clients’ privacy. Moreover, directed contact allows clients to choose their neighbors flexibly, meaning that the shared part model has a larger feature search space among clients, which guarantees better performance in a computation-constrained and communication-constrained scenario.

Theoretically, we present the non-trivially convergency analysis for the DFedPGP algorithm (see Section 4), which achieves a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$ in the general

non-convex setting. Empirically, we conduct extensive experiments on the CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets in non-IID settings with different data partitions. Experimental results confirm that the proposed algorithm can achieve competitive performance relative to other SOTA baselines (see Section 5) in PFL.

In summary, our main contributions are four-fold:

- We introduce the directed Push-sum optimization to PFL, which allows clients to choose their neighbors flexibly and guarantees a larger feature search space in a communication, and computation heterogeneity scenario.
- We propose a DPFL framework DFedPGP, incorporated with stochastic gradient push and partial model personalization for robust communication and fast convergence.
- We provide convergence guarantees for DFedPGP in the general non-convex setting with peer-to-peer partial participation in DPFL.
- Empirical results indicate the superiority of the proposed DFedPGP compared with various SOTA baselines and it can be well adapted to the data heterogeneous and computation resources constrained settings.

2. Related Work

Personalized Federated Learning (PFL). The PFL aims to produce the greatest personalized models for each client by model decoupling [1, 13], knowledge distillation [32, 39], multi-task learning [22, 54], model interpolation [15, 16] and clustering [17, 50]. More details can be referred to in [58]. In this paper, we mainly focus on the model decoupling methods, which divide the model into a global shared part and a personalized part, also called *partial personalization*. Existing partial personalized works in CFL achieve better performance than full model personalization with fewer shared parameters. FedPer [1], FedRep [13] and FedBABU [46] all use one global feature representation with many local classifiers but with differences in the relationship between the shared representation and the private linear parts. Fed-RoD [10] simultaneously trains a global full model and many private classifiers with both class-balanced loss and empirical loss. Theoretically, FedSim and FedAlt [47] provide the convergence analyses of both algorithms in the general non-convex setting, while FedAvg-P and Scaffold-P [12] improve the existing results in [47].

Decentralized Federated Learning (DFL). Due to the computation and communication resources heterogeneity among clients, DFL has been an encouraging field in recent years [5, 24, 33, 53], where clients only connect with their neighbors through peer-to-peer communication. We discuss the PFL methods in DFL considering multi-step local iterations. Specifically, DFedAvgM [56] applies multiple local iterations with SGD and the quantization method to reduce the communication cost. Dis-PFL [14] customizes the per-

sonalized model and pruned mask for each client to speed up the personalized convergence. KD-PDFL [23] leverages the knowledge distillation technique to empower each device to discern statistical distances between local models. ARDM [49] presents lower bounds on the communication and local computation costs for this personalized FL formulation in a peer-to-peer manner.

Push-sum over Directed Graphs. Push-sum optimizer is proposed to solve the asymmetric optimization problems over (time-varying) directed graphs. The first Push-sum study in [26] discusses gossip-type problems in directed graphs. PS-DDA [59] extends this method to a decentralized scenario and proves the convergence in a convex set. More optimization analysis can be referred to in [42, 44, 63–65]. As an effective optimizer, Push-sum and its variants have been applied to various machine learning (ML) tasks [2, 3, 11, 35, 57]. For example, SGP [2] combines Push-sum with stochastic gradient updates and also proposes the Overlap SGP, allowing overlaps of communication and computation to hide communication overhead. Quantized Push-sum [57] quantizes the Push-sum based algorithm over directed graphs to tackle the heavy communication load. AsyNG [11] proposes an asynchronous DFL system with directed communication by incorporating neighbor selection and gradient push to boost the performance on non-IID local data and heterogeneous edge nodes.

Nowadays, almost all PFL works suffer from the risk of deadlock from unstable communication channels and sub-optimal convergence from the different convergence-level aggregations. Therefore, we try to propose a framework of partial gradient push based on a directed communication graph for DPFL. It differs from the existing directed DFL methods in the exchange model part like OSGP[2], where clients focus on the whole parameters exchange for the only consensus model. Also, we adopt multi-step local steps and multiple alternate optimization for better convergence, which leads to an unbiased gradient estimation and the dependent stochastic variance between the shared parts and the personal parts. Therefore, the algorithm design and the theoretical analysis are both unique and non-trivial.

3. Methodology

In this section, we first define decentralized partial personalized models and the directed graph network in DPFL. Then we present the DFedPGP, which leverages the partial gradient push in the directed graph to mitigate the negative impact of heterogeneous data and computation resources.

3.1. Problem Setup

Decentralized Personalized Federated Learning. Consider a typical setting of DFL with m clients, where each

client i has the data distribution \mathcal{D}_i . We focus on the minimization of the finite sum of non-convex functions:

$$\min_{w \in \mathbb{R}^d} F(w) := \frac{1}{m} \sum_{i=1}^m F_i(w_i), \quad (1)$$

$$F_i(w_i) = \mathbb{E}_{\xi \sim \mathcal{D}_i} F_i(w_i; \xi).$$

where $F : \mathbb{R}^d \rightarrow \mathbb{R}$ is the global object function; $w_i \in \mathbb{R}^d$ represents the parameters of the machine learning model in client i ; F_i is the loss function associated with the data sample ξ randomly drawn from the distribution \mathcal{D}_i in client i .

To relieve the communication burden and improve personalized performance, we consider the partial model personalized version in DPFL. Specifically, the model parameters are partitioned into two parts: the *shared* parameters $u \in \mathbb{R}^{d_0}$ and the *personal* parameters $v_i \in \mathbb{R}^{d_i}$ for $i = 1, \dots, m$. The full model on client i is denoted as $w_i = (u_i, v_i)$. To simplify presentation, we denote $V = (v_1, \dots, v_m) \in \mathbb{R}^{d_1 + \dots + d_m}$, and then our goal is to solve this problem:

$$\min_{u, V} F(u, V) := \frac{1}{m} \sum_{i=1}^m F_i(u, v_i), \quad (2)$$

$$F_i(u, v_i) = \mathbb{E}_{\xi \sim \mathcal{D}_i} [F_i(u, v_i; \xi)].$$

where u denotes the consensus model averaged with u_i , i.e., $u = \frac{1}{m} \sum_{i=1}^m u_i$ and we use ∇_u and ∇_v to represent stochastic gradients with respect to u_i and v_i , respectively.

Directed Graph Network. In the decentralized network topology, the communication between clients can be modeled as a directed connected graph $\mathcal{G}(t) = (\mathcal{N}, \mathcal{V}(t), \mathcal{E}(t))$, where $\mathcal{N} = \{1, 2, \dots, m\}$ represents the set of clients, $\mathcal{V}(t) \subseteq \mathcal{N} \times \mathcal{N}$ represents the set of communication channels and $(i, j) \in \mathcal{E}(t)$ represents a directed link from client i to client j . Considering the time-varying directed graph, the link $(i, j) \in \mathcal{E}(t)$ (where $i \neq j$) does not imply the link $(j, i) \in \mathcal{E}(t)$. To further describe the directed communication, we define $N_i^{in} = \{j | (j, i) \in \mathcal{E}(t), j \in \mathcal{N}\}$ as the in-neighbor set and $N_i^{out} = \{j | (i, j) \in \mathcal{E}(t), j \in \mathcal{N}\}$ as the out-neighbor set, which are the sets with in-coming and out-coming links into node i separately.

Most works in DPFL assume the communication is based on a time-varying undirected graph, which satisfies $N_i^{in} = N_i^{out}$ and the link $(i, j) \in \mathcal{E}(t)$ (where $i \neq j$) must be equal to the link $(j, i) \in \mathcal{E}(t)$. But in reality, the undirected communication graph requires high attention in the implementation to avoid deadlocks. Directed communication graph networks mitigate this issue by flexibly selecting neighbors within clients and exhibiting higher robustness in terms of network communication quality.

3.2. Algorithm

In this section, DFedPGP (see Algorithm 1) is proposed to solve the problem (2) in a fully decentralized manner.

Partial Model Personalization. Drawing from previous research on CNNs, layers that serve specific engineering purposes: lower convolution layers (close to the input) are responsible for feature extraction, and the upper linear layers (close to the output) focus on complex pattern recognition [47]. The feature extraction layers, mapping data from high-dimensional feature space to an easily distinguished low space, are similar between clients but prone to overfitting. The linear classification layers, which determine the data category from the output of the previous feature extraction layers, are very different from data heterogeneity clients [31]. Therefore, we set the feature extraction layers as the shared parts and the linear classification layers as the personalized parts as [1, 13, 46, 47], and we leverage the alternating update approach for model training in Line 5-12, which aims to increase the compatibility between the personalized and the shared parts.

Push-sum Based DFedPGP. The Push-sum method [43] to solve the decentralized optimization problem performs one local stochastic gradient descent update with one iteration of push-pull transmission at each client. It maintains four variables locally at each round t : the biased shared model parts parameters u_i^t , the private model parts parameters v_i^t , the Push-sum bias weight μ_i^t , and the de-biased shared model parts parameters $z_i^t = u_i^t / \mu_i^t$. To save the overall communication, we introduce an idea from local SGD to perform a few epochs of local training before weights transmission. So at each round, every client performs a few local SGD steps in Lines 5–12 followed by one step of push-pull transmission in Lines 14–17. Notably, the local gradient is calculated at the de-biased parameters z_i^t in line 6 and they are then used to be updated in Line 10. The push-pull transmission includes the biased shared model parameters u_i^t and the Push-sum bias weight μ_i^t .

Directed Communication Graph. We set the mixing matrix P^t to describe the communication topology at each round t . DFedPGP can be adapted to various communication topologies such as time-varying, asymmetric, and sparse networks. We used the time-varying, asymmetric network here to encounter the limited communication bandwidth. Clients only need to know the outgoing mixing weights at each communication round and can independently choose the mixing weights from the other clients in the network. In this work, we introduce a simple yet effective random client selection method that satisfies our theory (Section 4) and the limited communication bandwidth in the experiments (Section 5).

4. Theoretical Analysis

In this section, we provide a detailed convergence theorem for the proposed algorithm DFedPGP and explore how the

Algorithm 1: DFedPGP

Input : Total number of devices m , total number of communication rounds T , local learning rate η_u and η_v , total number of local iterates K_u and K_v and mixing weight $p_{j,i}^t = 1/|\mathcal{N}_{i,0}^{out}|$.

Output: Personalized model $u_i^T = z_i^T$ and v_i^T after the final communication of all clients.

- 1 **Initialization:** Randomly initialize each device's shared parameters u_i^0 , the de-biased shared parameters $z_i^0 = u_i^0$, personal parameters v_i^0 and push-sum weight $\mu_i^0 = 1$.
- 2 **for** $t = 0$ **to** $T - 1$ **do**
- 3 **for** *client* i **in parallel** **do**
- 4 Set $u_i^{t,0} \leftarrow u_i^t$ and sample a batch of local data ξ_i and calculate local gradient iteration.
- 5 **for** $k = 0$ **to** $K_v - 1$ **do**
- 6 Perform personal parameters v_i update:

$$v_i^{t,k+1} = v_i^{t,k} - \eta_v \nabla_v F_i(z_i^{t,0}, v_i^{t,k}; \xi_i).$$
- 7 **end**
- 8 $v_i^{t+1} \leftarrow v_i^{t,K_v}$.
- 9 **for** $k = 0$ **to** $K_u - 1$ **do**
- 10 Update shared parameters u_i via

$$u_i^{t,k+1} = u_i^{t,k} - \eta_u \nabla_u F_i(z_i^{t,k}, v_i^{t+1}; \xi_i).$$

$$z_i^{t,k+1} = u_i^{t,k+1} / \mu_i^t.$$
- 11 **end**
- 12 $u_i^{t+1/2} \leftarrow u_i^{t,K_u}$.
- 13 Push weights $p_{j,i}^t u_i^{t+1/2}$ and bias information $p_{j,i}^t \mu_i^t$ to clients $j \in \mathcal{N}_{i,t}^{out}$.
- 14 Pull weights $p_{i,j}^t u_j^{t+1/2}$ and bias information $p_{i,j}^t \mu_j^t$ from clients $j \in \mathcal{N}_{i,t}^{in}$.
- 15 P2P updating by $u_i^{t+1} = \sum_{j \in \mathcal{N}_{i,t}^{in}} p_{i,j}^t u_j^{t+1/2}$ and

$$\mu_i^{t+1} = \sum_{j \in \mathcal{N}_{i,t}^{in}} p_{i,j}^t \mu_j^t.$$
- 16 De-bias the updated model by

$$z_i^{t+1} = u_i^{t+1} / \mu_i^{t+1}.$$
- 17 **end**
- 18 **end**
- 19 **end**

partial personalization and gradient push work. The detailed derivation process can be found in the appendix D.

4.1. Assumption

Assumption 1 (\mathcal{B} -bounded Connectivity [11]). The time-varying graph (i.e., the communication topology) is \mathcal{B} -bounded strongly connected to ensure the convergence of model training [2]. There exists a window size $\mathcal{B} \geq 1$ such that the graph union $\bigcup_{k=l}^{l+\mathcal{B}-1} \mathcal{G}(k)$ ($l = 0, 1, 2, \dots$) is strongly connected. Note that if $\mathcal{B} = 1$, each instance of graph $\mathcal{G}(k)$ is strongly connected at global iteration k .

Assumption 2 (Smoothness [47]). For each client $i = \{1, \dots, m\}$, the function F_i is continuously differentiable. There exist constants L_u, L_v, L_{uv}, L_{vu} such that for each client $i = \{1, \dots, m\}$:

- $\nabla_u F_i(u_i, v_i)$ is L_u -Lipschitz with respect to u_i and L_{uv} -Lipschitz with respect to v_i
- $\nabla_v F_i(u_i, v_i)$ is L_v -Lipschitz with respect to v_i and L_{vu} -Lipschitz with respect to u_i .

We summarize the relative cross-sensitivity of $\nabla_u F_i$ with respect to v_i and $\nabla_v F_i$ with respect to u with the scalar

$$\chi := \max\{L_{uv}, L_{vu}\}/\sqrt{L_u L_v}.$$

Assumption 3 (Bounded Variance [47]). The stochastic gradients in Algorithm 1 have bounded variance. That is, for all u_i and v_i , there exist constants σ_u and σ_v such that

$$\begin{aligned} \mathbb{E}[\|\nabla_u F_i(u_i, v_i; \xi_i) - \nabla_u F_i(u_i, v_i)\|^2] &\leq \sigma_u^2, \\ \mathbb{E}[\|\nabla_v F_i(u_i, v_i; \xi_i) - \nabla_v F_i(u_i, v_i)\|^2] &\leq \sigma_v^2. \end{aligned}$$

Assumption 4 (Partial Gradient Diversity [47]). There exist a constant σ_g^2 such that

$$\|\nabla_u F_i(u_i, v_i) - \nabla_u F(u_i, V)\|^2 \leq \sigma_g^2, \quad \forall u_i, V.$$

Assumption 1 is commonly adopted in gradient push work [11, 35, 42]: it is considerably weaker than requiring each $\mathcal{G}(t)$ be connected for it allows each client to connect in time-varying and directed topologies. Assumption 2–4 are mild and commonly used in characterizing the convergence rate of FL [25, 34, 52, 56].

4.2. Challenge and Proof

Challenges of Convergence Analysis. (1) Compared to the classical Push-sum based method like SGP[2], $u_i^{t,k} - u_i^{t,0}$ after multiple local iterations and alternately update is not an unbiased estimate of $\nabla F_i(u_i^t)$. Multiple local iteration analyses are non-trivial; (2) In contrast to the symmetric topology, DFedPGP communicates with its clients based on an asymmetric topology, resulting in $\sum_j p_{i,j}^t \neq 1$. As a consequence, each client needs to maintain a Push-sum weight μ_i^t to de-bias the model parameters; (3) To realize better-personalized performance, we need to analyze the convergence in a partial model personalized way, where the shared part u is updated with gradient pushing and pulling while the personalized part v is updated with local SGD separately. Now, we present the rigorous convergence analysis of DFedPGP as follows.

Theorem 1. *Under Assumptions 1-5, the local learning rates satisfy $0 < \eta_u < \frac{\delta}{4\sqrt{2}L_u K_u}$, F^* is denoted as the minimal value of F , i.e., $F(\bar{u}, V) \geq F^*$ for all $\bar{u} \in \mathbb{R}^d$, and $V = (v_1, \dots, v_m) \in \mathbb{R}^{d_1 + \dots + d_m}$. Let $\bar{u}^t = \frac{1}{m} \sum_{i=1}^m u_i^t$ and denote $\Delta_{\bar{u}}^t$ and Δ_v^t as:*

$$\Delta_{\bar{u}}^t = \|\nabla_u F(\bar{u}^t, V^{t+1})\|^2, \quad \Delta_v^t = \frac{1}{m} \sum_{i=1}^m \|\nabla_v F_i(u_i^t, v_i^t)\|^2.$$

Therefore, we have the convergence analysis below:

$$\begin{aligned} \frac{1}{T} \sum_{i=1}^T \left(\frac{1}{L_u} \mathbb{E}[\Delta_{\bar{u}}^t] + \frac{1}{L_v} \mathbb{E}[\Delta_v^t] \right) &\leq \mathcal{O} \left(\frac{F(\bar{u}^1, V^1) - F^*}{\sqrt{T}} \right. \\ &+ \frac{(1+L_v)\sigma_v^2}{\sqrt{T}} + (\sigma_u^2 + \sigma_g^2) \left(\frac{C^2}{(1-q)^2 L_u T} + \frac{1}{K_u L_u \sqrt{T}} \right) \\ &+ \left. \frac{1}{K_u L_u \delta^2 T^{3/2}} + \frac{C^2}{K_u L_u (1-q)^2 T^{3/2}} + \frac{L_{vu}^2 C^2}{(1-q)^2 L_u^2 \sqrt{T}} \right). \end{aligned} \quad (3)$$

The parameters C and q are related to the communication topology as [2, Lemma 3]. δ is the minimum sum of any row elements in the matrix $\prod_{i=1}^t \mathcal{G}(i)$ for all $t \geq 0$ as [57, Proposition 2.1]. With the proper step sizes, we have the following corollary.

Corollary 1 (Convergence Rate for DFedPGP). *Under Theorem 1 and by setting the local learning rates $\eta_u = \mathcal{O}(1/L_u K_u \sqrt{T})$, $\eta_v = \mathcal{O}(1/L_v K_v \sqrt{T})$, it holds that:*

$$\begin{aligned} \frac{1}{T} \sum_{i=1}^T \left(\frac{1}{L_u} \mathbb{E}[\Delta_{\bar{u}}^t] + \frac{1}{L_v} \mathbb{E}[\Delta_v^t] \right) \\ \leq \mathcal{O} \left(\frac{F(\bar{u}^1, V^1) - F^* + \sigma_1^2}{\sqrt{T}} + \frac{\sigma_2^2}{T} + \frac{\sigma_3^2}{\sqrt{T^3}} \right), \end{aligned} \quad (4)$$

where

$$\begin{aligned} \sigma_1^2 &= (1+L_v)\sigma_v^2 + \left(\frac{1}{K_u L_u} + \frac{L_v \chi^2 C^2}{(1-q)^2 L_u} \right) (\sigma_u^2 + \sigma_g^2), \\ \sigma_2^2 &= \frac{C^2}{(1-q)^2 L_u} (\sigma_u^2 + \sigma_g^2), \\ \sigma_3^2 &= \left(\frac{1}{K_u L_u \delta^2} + \frac{C^2}{(1-q)^2 K_u L_u} \right) (\sigma_u^2 + \sigma_g^2). \end{aligned} \quad (5)$$

Remark 1. Corollary 1 provides explicit insights into how various key factors affect the convergence of DFedPGP. Specifically, the convergence analysis illustrates that the large values of the gradient variance σ_u^2 , σ_v^2 , σ_g^2 and gradient bounded B lead to slower convergence. It also shows that more local update steps K_u accelerate the convergence, quantitatively justifying the benefit of exploiting more local updates in the algorithm. Also, the smoothness of local loss functions such as L_u , L_v , and L_{vu} , have a significant influence on the convergence bound.

Remark 2. As the definition in [2], the q in Corollary 1 can be explicitly expressed as $q = (1 - a^{\Delta \mathcal{B}})^{\frac{1}{\Delta \mathcal{B} + 1}}$, where Δ is the diameter of the communication network, \mathcal{B} is the same defined in Assumption 1, and $a < 1$ is a constant. Note that the bound will be tighter as q decreases, which means the network connectivity improves and clients exchange parameters with more neighbors in the communication progress. Moreover, the connectivity constant C decreases as the connectivity of the communication network improves, which leads to the same conclusion as q . More details about the communication constant can be found in Lemma 3.

Table 1. Test accuracy (%) on CIFAR-10 & 100 in both Dirichlet and Pathological distribution settings.

| Algorithm | CIFAR-10 | | | | CIFAR-100 | | | |
|-----------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| | Dirichlet | | Pathological | | Dirichlet | | Pathological | |
| | $\alpha = 0.1$ | $\alpha = 0.3$ | $c = 2$ | $c = 5$ | $\alpha = 0.1$ | $\alpha = 0.3$ | $c = 5$ | $c = 10$ |
| Local | 78.96 \pm .42 | 63.20 \pm .28 | 85.16 \pm .18 | 68.56 \pm .35 | 39.38 \pm .33 | 22.59 \pm .49 | 71.34 \pm .46 | 53.15 \pm .31 |
| FedAvg | 84.17 \pm .28 | 79.66 \pm .20 | 85.04 \pm .11 | 81.18 \pm .27 | 57.43 \pm .03 | 55.06 \pm .06 | 69.05 \pm .43 | 66.37 \pm .48 |
| FedPer | 88.57 \pm .09 | 84.06 \pm .29 | 90.94 \pm .24 | 86.97 \pm .35 | 54.23 \pm .14 | 34.07 \pm .76 | 78.48 \pm .93 | 70.38 \pm .02 |
| FedRep | 88.78 \pm .40 | 84.50 \pm .05 | 91.09 \pm .12 | 86.22 \pm .51 | 44.02 \pm .98 | 26.88 \pm .49 | 78.77 \pm .19 | 67.65 \pm .43 |
| FedBABU | 87.79 \pm .53 | 83.26 \pm .09 | 91.28 \pm .15 | 83.90 \pm .24 | 60.23 \pm .07 | 52.37 \pm .82 | 77.50 \pm .33 | 69.81 \pm .12 |
| Ditto | 80.22 \pm .10 | 73.51 \pm .04 | 84.96 \pm .40 | 75.59 \pm .32 | 48.85 \pm .54 | 48.65 \pm .50 | 69.48 \pm .45 | 60.77 \pm .30 |
| DFedAvgM | 86.94 \pm .62 | 82.49 \pm .57 | 90.23 \pm .97 | 85.26 \pm .47 | 58.80 \pm .82 | 54.89 \pm .77 | 75.89 \pm .65 | 70.55 \pm .44 |
| OSGP | 87.39 \pm .13 | 83.14 \pm .18 | 90.72 \pm .08 | 84.69 \pm .25 | 59.76 \pm .69 | 54.98 \pm .48 | 76.70 \pm .59 | 71.08 \pm .52 |
| Dis-PFL | 87.77 \pm .46 | 82.71 \pm .28 | 88.19 \pm .47 | 84.18 \pm .61 | 56.06 \pm .20 | 46.65 \pm .18 | 71.79 \pm .42 | 65.35 \pm .10 |
| DFedPGP | 88.85 \pm .21 | 85.61 \pm .05 | 91.26 \pm .05 | 87.12 \pm .37 | 66.26 \pm .25 | 57.66 \pm .42 | 78.78 \pm .41 | 72.19 \pm .21 |

Remark 3. From Corollary 1, the proposed DFedPGP has a convergence rate of $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$. This result is consistent with the convergence rate achieved by [47, 52] in PFL. Moreover, when the smoothness of the shared parameters is not good, it means that L_u is large, the term $\mathcal{O}\left(\frac{1}{T} + \frac{1}{\sqrt{T^3}}\right)$ can be neglected compared to $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$.

5. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of the proposed DFedPGP in data heterogeneity and computation resources heterogeneity scenarios. Below, we first introduce the experimental setup.

5.1. Experiment Setup

Dataset and Data Partition. We evaluate the performance on CIFAR-10, CIFAR-100 [27], and Tiny-ImageNet [29] datasets in the Dirichlet distribution and Pathological distribution, where CIFAR-10 and CIFAR-100 are two real-life image classification datasets with total 10 and 100 classes. Experiments on the Tiny-ImageNet dataset are placed in Appendix C.3 due to the limited space. We partition the training and testing data according to the same Dirichlet distribution $\text{Dir}(\alpha)$ such as $\alpha = 0.1$ and $\alpha = 0.3$ for each client. The smaller the α is, the more heterogeneous the setting is. Meanwhile, for each client, we sample 2 and 5 classes from a total of 10 classes on CIFAR-10, and 5 and 10 classes from a total of 100 classes on CIFAR-100, respectively [72]. The number of sampling classes is represented as “c” in Table 1 and the fewer classes each client owns, the more heterogeneous the setting is.

Baselines and Backbone. We compare the proposed methods with the SOTA baselines PFL. For instance, Local is the simplest method where each client only conducts training on their own data without communicating with other

clients. Federated learning methods include FedAvg [41], FedPer [1], FedRep [13], FedBABU [46] and Ditto [37]. For DFL methods, we take DFedAvgM [56], Dis-PFL [14] and OSGP [2] as our baselines. All methods are evaluated on ResNet-18 [20] and replace the batch normalization with the group normalization followed by [14, 53, 62] to avoid unstable performance. For the partial PFL methods, we set the lower linear classifier layers as the personal part responsible for complex pattern recognition, and the rest upper representation layer as the shared layers focusing on feature extraction. Note that we compare the personal test accuracy for all methods since our goal is to solve PFL.

Implementation Details. We keep the same experiment setting for all baselines and perform 500 communication rounds with 100 clients. The client sampling ratio is 0.1 in CFL, while each client communicates with 10 neighbors in PFL accordingly. The batch size is 128. For DFedPGP, we train the shared part for 5 epochs per round as the same as other baselines, and train 1 epoch for the personal part to align the shared part and save the computation consumption. We set SGD [48] as the base optimizer for all methods with a learning rate $\eta_u = 0.1$ to update the model parameters and the learning rate decreasing by 0.99 \times exponentially. All methods are set with a decay rate of 0.005 and a local momentum of 0.9. We report the mean performance with 3 different random seeds and more details of the baseline methods can be found in Appendix C.1.

5.2. Performance Evaluation

Comparison with the Baselines. As shown in Table 1 and Figure 2, the proposed DFedPGP outperforms other baseline methods with the best stability and better performance in both different datasets and different data heterogeneity scenarios. Specifically, on the CIFAR-10 dataset, DFedPGP achieves 86.50% on the Dirichlet-0.3 setups, 1.11% ahead of the best-comparing method FedRep. On the CIFAR-

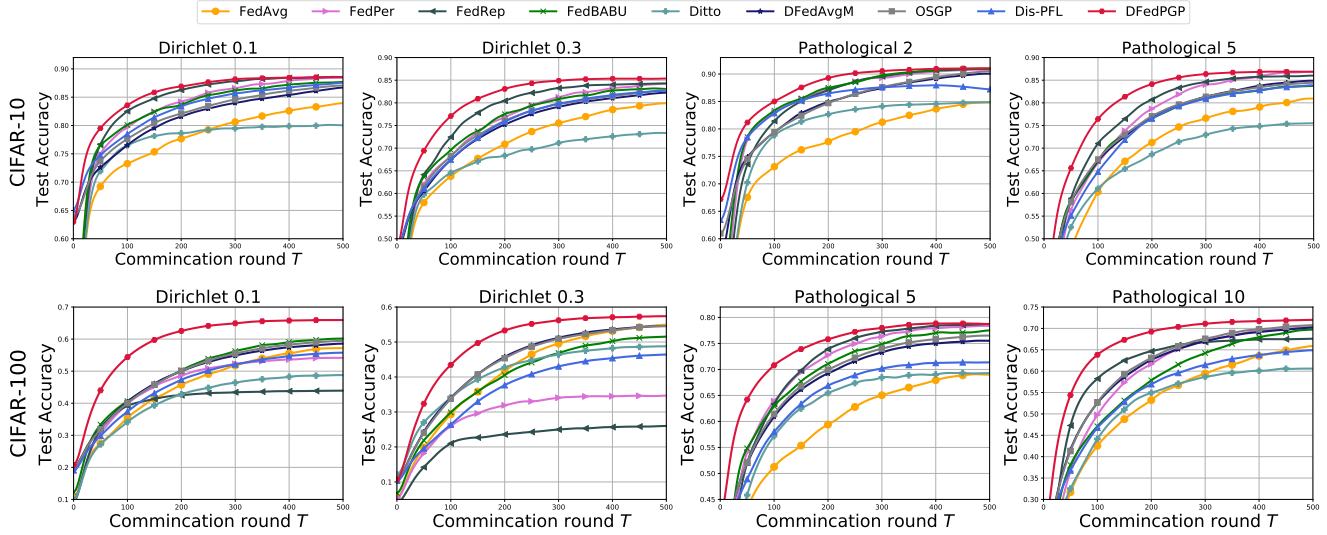


Figure 2. Test accuracy on CIFAR-10 (first line) and CIFAR-100 (second line) with heterogenous data partitions. With limited pages, we only show the training progress of the typical methods.

100 dataset, DFedPGP achieves at least 2.60% and 1.11% improvement from the other baselines on the Directlet-0.3 and Pathological-10 settings. The communication based on the directed graph allows clients to choose their in-neighbors and out-neighbors flexibly, guaranteeing that they can choose useful information for their local training.

Comparison on Heterogeneous Setting. We discuss two data heterogeneities, Dirichlet distribution and Pathological distribution in Table 1, and prove the effectiveness and robustness of the DFedPGP. In PFL tasks, since the local training can't cater for all classes inside clients, the accuracy decreases with the heterogeneity decreasing. ¹ On the CIFAR-10 dataset, when the heterogeneity decreases from 0.1 to 0.3 in Directlet distribution, FedRep drops from 88.78% to 84.50%, while DFedPGP drops about 3.24% to 85.61%, meaning its stronger stability for several heterogeneous settings. On the Pathological distribution, DFedPGP beat the best-compared baselines over 1.11% on the CIFAR-100 dataset with only 10 categories per client, which confirm that the proposed methods could achieve better performance in the strong heterogeneity.

Comparison on the Convergence Speed with Baselines. We show the convergence speed via the learning curves of the compared methods in Figure 2 and Table 2. DFedPGP

¹Generally, the higher data heterogeneity means a greater difference between local data distribution. But in extreme data heterogeneity PFL tasks, the higher heterogeneity means it owns fewer data classes locally, which makes the classification task easier and clients will achieve better performance. For example, in the Pat-2 setting, the local binary classification task is easier than the five classification tasks in the Pat-5 setting, so the average test performance in Pat-2 is better than that in Pat-5. The same phenomenon can be seen in most PFL works [14, 22, 46, 66, 72].

achieves the fastest convergence speed among the comparison methods, which benefits from the direct partial model transmission and alternate update. For example, DFedPGP is almost twice as fast as the other methods in Dirichlet-0.3 on CIFAR-10 and CIFAR-100 settings. In comparison with the CFL methods, directly learning the neighbors' feature representation in DFL can speed up the convergence rate for personalized problems. Notably, we target the setting where the busiest node's communication bandwidth is restricted for fairness when compared with the CFL methods.

Comparison on Computation Resources Heterogeneity.

In reality, the federating process often involves heterogeneous devices, which means the shared parameters are trained at different convergence levels. We follow [4, 14] to

Table 2. The required communication rounds when achieving the target accuracy (%).

| Algorithm | CIFAR-10 | | CIFAR-100 | |
|----------------|------------|------------|------------|------------|
| | Dir-0.3 | Pat-2 | Dir-0.3 | Pat-10 |
| | acc@80 | acc@90 | acc@45 | acc@65 |
| FedAvg | - | - | 234 | 456 |
| FedPer | 262 | 343 | - | 246 |
| FedRep | 189 | 322 | - | 210 |
| FedBABU | 270 | 312 | 261 | 314 |
| Ditto | - | - | 279 | - |
| DFedAvgM | 320 | 452 | 187 | 249 |
| OSGP | 309 | 439 | 192 | 230 |
| Dis-PFL | 307 | - | 368 | 492 |
| DFedPGP | 131 | 224 | 111 | 113 |

Table 3. Test accuracy (%) in computation resources heterogeneous setting.

| Algorithm | FedAvg | FedPer | FedRep | FedBABU | Ditto | DFedAvgM | Dis-PFL | OSGP | DFedPGP |
|-----------|--------|--------|--------|---------|-------|----------|---------|-------|--------------|
| Dir | 75.76 | 81.06 | 83.08 | 72.66 | 75.63 | 82.70 | 82.41 | 82.81 | 83.63 |
| Pat | 81.04 | 91.09 | 88.57 | 83.06 | 82.26 | 91.52 | 91.40 | 91.61 | 91.83 |

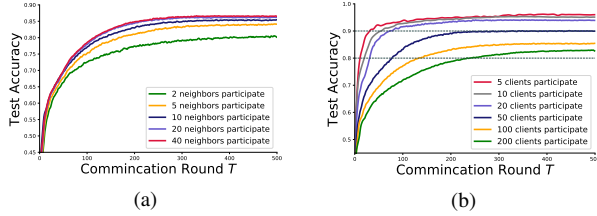


Figure 3. Ablation study. (a) Effect of the number of neighboring clients. (b) Effect of the number of participated clients.

divide 100 clients into 5 parts and transmit their parameters after 1, 2, 3, 4 and 5 local epochs to simulate the different computation capabilities of each device [60].

Table 3 shows the comparison among PFL methods under a computation heterogeneous setting on the CIFAR10 dataset with Dirichlet-0.3 distribution. DFedPGP achieves the best compared with the other baselines, indicating that the partial gradient push can alleviate the effect of the different convergence level aggregation. Another interesting finding is that FedRep is the best PFL method encountering the computation heterogeneous challenge in CFL, indicating that keeping the classifiers locally and updating the private and shared parts alternately is an effective way to solve the computation heterogeneity problem.

5.3. Ablation Study

Number of Neighboring Clients. We conduct experiments on the convergence performance under different neighbor participation numbers of {2, 5, 10, 20, 40} on CIFAR-10 with Dir-0.3 distribution. As shown in Figure 3a, the highest personalized performance is achieved when the participation number is set to 40, which indicates that with more clients exchanging their information, a quicker convergence speed will be achieved, aligning with our insight. Notably, the proposed DFedPGP can realize a stable convergence even when transmitting information to only 2 neighbors.

Number of Participated Clients. As depicted in Figure 3b, we compare the personalized performance between different numbers of client participation of {5, 10, 20, 50, 100, 200} on the CIFAR-10 dataset with Dirichlet-0.3 distribution under the same hyper-parameters. Compared with larger participated clients {50, 100, 200}, the smaller participated clients {5, 10} can achieve better test accuracy and convergence as the number of local data increases.

Module Augmentation Ablation. We investigate the ef-

Table 4. Test accuracy (%) of different module augmentation.

| Algorithm | Partial Personalization | Directed Communication | Dir | Pat |
|------------|-------------------------|------------------------|--------------|--------------|
| DFedAvgM | | | 82.49 | 90.23 |
| DFedAvgM-P | ✓ | | 84.69 | 90.90 |
| OSGP | | ✓ | 83.14 | 90.72 |
| DFedPGP | ✓ | ✓ | 85.61 | 91.26 |

fect of partial personalization and directed communication with different data heterogeneity on the CIFAR-10 dataset. From Table 4, DFedPGP achieves the best in both Dirichlet-0.3 and Pathological-2. In the comparison of partial personalization, DFedAvgM-P and DfedPGP outperform their full personalization versions DFedAvgM and OSGP by a great margin separately. In directed communication comparison, DfedPGP and OSGP outperform their undirected versions DFedAvgM-P and DFedAvgM, respectively. From the ablation study, both partial personalization and directed communication have a great influence on decentralized and personalized performance. Randomly choosing clients’ in-neighbors and out-neighbors in directed graphs means that the shared part model has a larger feature search space among clients, compared with the undirected graphs. Intuitively, this increases the involved clients in one communication round and enhances communication efficiency.

6. Conclusion

In this paper, we propose a novel method DFedPGP for PFL, which simultaneously guarantees robust communication and better personalized performance with convergence guarantee via partial gradient push in a directed communication graph. The directed collaboration allows clients to choose their corporate neighbors flexibly, which guarantees effective aggregation and learning under data and device heterogeneity scenarios. For theoretical findings, we present the personalized convergence rate of $\mathcal{O}(1/\sqrt{T})$ in the non-convex setting for DFedPGP. Empirical results also verify the superiority of the proposed approach.

Future Works. In the current work, we mainly focus on the theoretical analysis and the experiment verification of the partial Push-sum based optimization framework with the directed graph for DPFL. It can be extended with effective client selection methods to speed up the convergence and improve personalized performance in the future.

Acknowledgment. This work is supported by STI 2030—Major Projects (No. 2021ZD0201405).

References

- [1] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. [1](#), [2](#), [4](#), [6](#), [13](#)
- [2] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353. PMLR, 2019. [3](#), [4](#), [5](#), [6](#), [13](#)
- [3] Mahmoud S Assran and Michael G Rabbat. Asynchronous gradient push. *IEEE Transactions on Automatic Control*, 66(1):168–183, 2020. [3](#), [17](#)
- [4] Dmitrii Avdiukhin and Shiva Kasiviswanathan. Federated learning under arbitrary communication patterns. In *International Conference on Machine Learning*, pages 425–435. PMLR, 2021. [7](#)
- [5] 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. *arXiv preprint arXiv:2211.08413*, 2022. [2](#), [12](#)
- [6] Michael Blot, David Picard, Matthieu Cord, and Nicolas Thome. Gossip training for deep learning. *arXiv preprint arXiv:1611.09726*, 2016. [12](#)
- [7] Zheng Chai, Hannan Fayyaz, Zeshan Fayyaz, Ali Anwar, Yi Zhou, Nathalie Baracaldo, Heiko Ludwig, and Yue Cheng. Towards taming the resource and data heterogeneity in federated learning. In *2019 USENIX conference on operational machine learning (OpML 19)*, pages 19–21, 2019. [2](#)
- [8] Daoyuan Chen, Dawei Gao, Yuexiang Xie, Xuchen Pan, Zitao Li, Yaliang Li, Bolin Ding, and Jingren Zhou. Fs-real: Towards real-world cross-device federated learning. *arXiv preprint arXiv:2303.13363*, 2023.
- [9] Daoyuan Chen, Liuyi Yao, Dawei Gao, Bolin Ding, and Yaliang Li. Efficient personalized federated learning via sparse model-adaptation. *arXiv preprint arXiv:2305.02776*, 2023. [2](#)
- [10] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. *arXiv preprint arXiv:2107.00778*, 2021. [2](#)
- [11] Min Chen, Yang Xu, Hongli Xu, and Liusheng Huang. Enhancing decentralized federated learning for non-iid data on heterogeneous devices. pages 2289–2302, 2023. [2](#), [3](#), [4](#), [5](#)
- [12] Yiming Chen, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Sharper convergence guarantees for federated learning with partial model personalization. *arXiv preprint arXiv:2309.17409*, 2023. [2](#)
- [13] 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. [2](#), [4](#), [6](#), [13](#)
- [14] Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. Dispf: Towards communication-efficient personalized federated learning via decentralized sparse training. In *International Conference on Machine Learning, ICML*, pages 4587–4604. PMLR, 2022. [1](#), [2](#), [6](#), [7](#), [13](#)
- [15] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020. [2](#)
- [16] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*, 2020. [2](#)
- [17] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020. [2](#)
- [18] Abolfazl Hashemi, Anish Acharya, Rudrajit Das, Haris Vikalo, Sujay Sanghavi, and Inderjit Dhillon. On the benefits of multiple gossip steps in communication-constrained decentralized federated learning. *IEEE Transactions on Parallel and Distributed Systems, TPDS*, pages 2727–2739, 2022. [12](#)
- [19] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33:14068–14080, 2020. [2](#)
- [20] 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. [6](#)
- [21] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020. [12](#)
- [22] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7865–7873, 2021. [1](#), [2](#), [7](#)
- [23] Eunjeong Jeong and Marios Kountouris. Personalized decentralized federated learning with knowledge distillation. *arXiv preprint arXiv:2302.12156*, 2023. [3](#)
- [24] Jiawen Kang, Dongdong Ye, Jiangtian Nie, Jiang Xiao, Xi-anjun Deng, Siming Wang, Zehui Xiong, Rong Yu, and Dusit Niyato. Blockchain-based federated learning for industrial metaverses: Incentive scheme with optimal aoi. In *2022 IEEE International Conference on Blockchain (Blockchain)*, pages 71–78. IEEE, 2022. [2](#)
- [25] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020. [5](#)
- [26] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE, 2003. [3](#), [12](#)
- [27] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. [6](#)
- [28] Anusha Lalitha, Shubhanshu Shekhar, Tara Javidi, and Fari-naz Koushanfar. Fully decentralized federated learning. In *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018. [12](#)

- [29] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6
- [30] Boyue Li, Shicong Cen, Yuxin Chen, and Yuejie Chi. Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. *Journal of Machine Learning Research, JMLR*, pages 180:1–180:51, 2020. 12
- [31] 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. 1, 4
- [32] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019. 2
- [33] Jun Li, Yumeng Shao, Kang Wei, Ming Ding, Chuan Ma, Long Shi, Zhu Han, and H. Vincent Poor. Blockchain assisted decentralized federated learning (blade-fl): Performance analysis and resource allocation. *IEEE Transactions on Parallel and Distributed Systems*, 33(10):2401–2415, 2022. 2
- [34] Qinglun Li, Li Shen, Guanghao Li, Qunjun Yin, and Dacheng Tao. Dfedadm: Dual constraints controlled model inconsistency for decentralized federated learning. *arXiv preprint arXiv:2308.08290*, 2023. 5
- [35] Qinglun Li, Miao Zhang, Nan Yin, Qunjun Yin, and Li Shen. Asymmetrically decentralized federated learning. *arXiv preprint arXiv:2310.05093*, 2023. 3, 5
- [36] Shuangtong Li, Tianyi Zhou, Xinmei Tian, and Dacheng Tao. Learning to collaborate in decentralized learning of personalized models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9766–9775, 2022. 2, 12
- [37] 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. 6, 13
- [38] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017. 12
- [39] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020. 1, 2
- [40] Tao Lin, Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. *arXiv preprint arXiv:2102.04761*, 2021. 12
- [41] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguerre y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 6, 13
- [42] Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2014. 3, 5
- [43] Angelia Nedić and Alex Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Transactions on Automatic Control*, 61(12):3936–3947, 2016. 4
- [44] Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017. 3
- [45] TV Nguyen, MA Dakka, SM Diakiw, MD VerMilyea, M Perugini, JMM Hall, and D Perugini. A novel decentralized federated learning approach to train on globally distributed, poor quality, and protected private medical data. *Scientific Reports*, 12(1):8888, 2022. 12
- [46] Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Towards enhanced representation for federated image classification. *arXiv preprint arXiv:2106.06042*, 2021. 1, 2, 4, 6, 7, 13
- [47] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. pages 17716–17758, 2022. 2, 4, 5, 6, 16
- [48] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951. 6
- [49] Abdurakhmon Sadiev, Ekaterina Borodich, Aleksandr Beznosikov, Darina Dvinskikh, Savelyi Chezhegov, Rachael Tappenden, Martin Takáč, and Alexander Gasnikov. Decentralized personalized federated learning: Lower bounds and optimal algorithm for all personalization modes. *EURO Journal on Computational Optimization*, 10:100041, 2022. 3
- [50] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020. 2
- [51] Eugene Seneta. *Non-negative matrices and Markov chains*. Springer Science & Business Media, 2006. 12
- [52] 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. *arXiv preprint arXiv:2305.15157*, 2023. 5, 6, 13
- [53] Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. Improving the model consistency of decentralized federated learning. *arXiv preprint arXiv:2302.04083*, 2023. 1, 2, 6
- [54] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019. 2
- [55] Yi Sui, Junfeng Wen, Yenson Lau, Brendan Leigh Ross, and Jesse C Cresswell. Find your friends: Personalized federated learning with the right collaborators. *arXiv preprint arXiv:2210.06597*, 2022. 1
- [56] Tao Sun, Dongsheng Li, and Bao Wang. Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2, 5, 6, 13

- [57] Hossein Taheri, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. Quantized decentralized stochastic learning over directed graphs. In *International Conference on Machine Learning*, pages 9324–9333. PMLR, 2020. [3](#), [5](#), [17](#)
- [58] Alysa Ziyang Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. [2](#)
- [59] Konstantinos I Tsianos, Sean Lawlor, and Michael G Rabbat. Push-sum distributed dual averaging for convex optimization. In *2012 IEEE 51st IEEE conference on decision and control (CDC)*, pages 5453–5458. IEEE, 2012. [3](#)
- [60] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020. [8](#)
- [61] Lun Wang, Yang Xu, Hongli Xu, Min Chen, and Liusheng Huang. Accelerating decentralized federated learning in heterogeneous edge computing. *IEEE Transactions on Mobile Computing*, 2022. [12](#)
- [62] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. [6](#)
- [63] Chenguang Xi and Usman A Khan. On the linear convergence of distributed optimization over directed graphs. *arXiv preprint arXiv:1510.02149*, 2015. [3](#)
- [64] Chenguang Xi and Usman A Khan. Dextra: A fast algorithm for optimization over directed graphs. *IEEE Transactions on Automatic Control*, 62(10):4980–4993, 2017.
- [65] Ran Xin and Usman A Khan. A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Systems Letters*, 2(3):315–320, 2018. [3](#)
- [66] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867*, 2023. [7](#)
- [67] Haishan Ye and Tong Zhang. Deepca: Decentralized exact pca with linear convergence rate. *J. Mach. Learn. Res.*, 22(238):1–27, 2021. [12](#)
- [68] Haishan Ye, Ziang Zhou, Luo Luo, and Tong Zhang. Decentralized accelerated proximal gradient descent. *Advances in Neural Information Processing Systems*, 33:18308–18317, 2020. [12](#)
- [69] Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023. [1](#)
- [70] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pages 7184–7193. PMLR, 2019. [2](#)
- [71] Zhengxin Yu, Jia Hu, Geyong Min, Han Xu, and Jed Mills. Proactive content caching for internet-of-vehicles based on peer-to-peer federated learning. In *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 601–608. IEEE, 2020. [12](#)
- [72] Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020. [6](#), [7](#)
- [73] Tongtian Zhu, Fengxiang He, Lan Zhang, Zhengyang Niu, Mingli Song, and Dacheng Tao. Topology-aware generalization of decentralized sgd. In *International Conference on Machine Learning, ICML*, pages 27479–27503. PMLR, 2022. [12](#)