Sparsified Random Partial Model Update for Personalized Federated Learning

Xinyi Hu, Zihan Chen, *Member, IEEE,* Chenyuan Feng, *Member, IEEE,* Geyong Min, *Senior Member, IEEE,* Tony Q. S. Quek, *Fellow, IEEE,* and Howard H. Yang, *Member, IEEE*

***Abstract*—Federated Learning (FL) stands as a privacy-preserving machine learning paradigm that enables collaborative training of a global model across multiple clients. However, the practical implementation of FL models often confronts challenges arising from data heterogeneity and limited communication resources. To address the aforementioned issues simultaneously, we develop a Sparsified Random Partial Update framework for personalized Federated Learning (`SRP-pFed`), which builds upon the foundation of dynamic partial model updates. Specifically, we decouple the local model into personal and shared parts to achieve personalization. For each client, the ratio of its personal part associated with the local model, referred to as the update rate, is regularly renewed over the training procedure via a random walk process endowed with reinforced memory. In each global iteration, clients are clustered into different groups where the ones in the same group share a common update rate. Benefiting from such design, `SRP-pFed` realizes model personalization while substantially reducing communication costs in the uplink transmissions. We conduct extensive experiments on various training tasks with diverse heterogeneous data settings. The results demonstrate that the `SRP-pFed` consistently outperforms the state-of-the-art methods in test accuracy and communication efficiency.**

***Index Terms*—Personalized federated learning, sparsification, client clustering, convergence rate.**

I. Introduction

THE surge in edge devices, including smartphones and Internet-of-Things (IoT) devices, each equipped with abundant sensing, computation, and storage resources, results in substantial daily data generation at the network edge. This data can be harnessed for machine learning models, facilitating various intelligent services, ranging from personal fitness tracking [1], traffic monitoring [2], to smart home security [3]. Traditional machine learning approaches involve transferring all the raw data to the cloud for model training, incurring high communication costs and posing severe privacy risks.

In contrast, federated learning stands as an alternative machine learning paradigm that allows multiple clients to train a shared model collaboratively without divulging their local private data. This contributes to the realization of trustworthy edge intelligent systems. Nevertheless, implementing federated learning in practice presents pervasive challenges. In particular, datasets possessed by different clients are, by nature, heterogeneous, exhibiting highly non-independently and identically distributed (non-IID) features. Besides, the clients' datasets vary vastly in the amount of data samples [4]. Such discrepancies in the distribution and sizes of clients' local datasets are commonly known as data heterogeneity, imposing crucial challenges to the convergence and stability performance of the FL training. On the other hand, although the ever-increasing processing capabilities of end-user devices promote the deployment of large Deep Neural Networks (DNNs) at the edge entities, the hefty communication overhead incurred by the frequent exchange of models between clients and server hinders the scalability of FL systems.

In response, numerous methods have been proposed to address these two critical issues [5]–[8]. Among them, personalized FL (PFL) is especially effective in coping with the constraint of a single global model in conventional FL, which has the setback of restricting the generalization capability of the FL model into heterogeneous local data. In contrast to the conventional FL training scheme, PFL seeks to train personalized models for every (or a group of) client(s) with similar preferences. This is achieved by applying different learning paradigms in the FL setting.

Personalization techniques can be categorized into similarity-based and architecture-based approaches [4]. The former achieves personalization by modeling client relationships, while the latter provides a personalized model architecture tailored to each client. Specifically, similarity-based approaches for personalizing FL models involve extracting a shared model based on similarities among client relationships and training multiple local models, such as multi-task learning [9], model interpolation [10], and clustering-based hierarchical framework [11]. However, most of these approaches still have to exchange the full model, resulting in a large communication overhead.

To enhance communication efficiency with concurrent personalization, architecture-based PFL approaches have been proposed, which decouple each client's private weights from the globally shared ones in the local model [4]. These private weights are trained locally only on the devices and not shared with the server. This weight decoupling method is often used in conjunction with classical DNN pruning algorithms, e.g., structured pruning, where weights are grouped in different fine-grained structural units, and each structural unit will be assigned to the same part (private or shared). A pictorial example is given in Figure 1 (a) & (b), where each layer is treated as a structural unit. The authors of FedRep [12] separate the deep layer into private parts to learn personalized task-specific representations, while the shallow layers are shared with the server to learn low-level generic features. On the contrary, LG-Fed [13] shares the deep layers. Another approach is via channel-wise decoupling. As demonstrated in Figure 1 (c) and detailed by CD^2-pFed [14], this method achieves model personalization in both low-level and high-level representa-

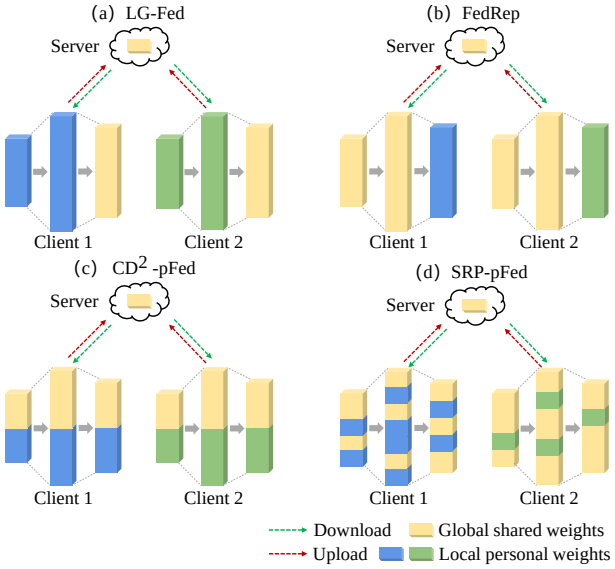

Fig. 1. A comparison of various partial model PFL approaches is illustrated. In (a), (b), and (c), the shared and private components of the model are manually divided, and all clients utilize the same update rate. In contrast, our proposed method depicted in (d) assigns distinct update rates to individual clients, with each client having a unique structural configuration.

tions, tackling the feature heterogeneity, distribution skew, and concept shift. There are also methods combined with unstructured pruning, for instance, the LotteryFL [15] and FedDST [16], that evaluate whether each weight is shared with the server separately. Nevertheless, in these schemes, the ratio between the private and the shared part, namely the *update rate*, is set empirically, and all clients adhere to a common one. Notably, the update rate controls the number of private weights for learning local representations, which affects the aggregated model's capability to learn good representations on heterogeneous data. The amount of global information required by the federated system varies in temporal and spatial dimensions, encompassing different communication rounds and distinct clients during the same communication rounds. As such, the empirical fixed update rates may be ineffective in finding the optimal personalization architectures, leading to poor performance.

In light of these challenges, we propose `SRP-pFed`, a Sparsified Random Partial Update framework for personalized Federated Learning, devised based on dynamic partial model update. As illustrated in Figure 1, different from previous personalization approaches that train the global model under constant update rates, the proposed approach provides an adaptive update rate allocation mechanism. Specifically, `SRP-pFed` refines the update rate with reinforced memory over the spatial and temporal dimensions along the model training process: In the spatial dimension, the clients are clustered into several groups, wherein each group shares a common update rate based on the local model weights. Recognizing that the weights change during the training process–hence, a one-shot clustering is not precise enough–we introduce a dynamic clustering-training loop in the time dimension. We sample K update rates with the probability of being sampled set according to the cluster results and local model performance in the previous round. In each loop, the update rate is constructed by a random walk process. Each client downloads these K classes of global shared weights and updates the corresponding elements of the local model while the rest of the element values remain unchanged. The updated model with the lowest loss of each client will participate in the subsequent partial model update process, in which only the shared weight will be uploaded to the server in this loop. The loop is executed repeatedly during the training process. This design involves uploading part of the model while downloading the complete model during training, addressing the bottleneck caused by asymmetric network speeds in the federated system. Specifically, the upload link (from client to server) is usually slower than the download link (from server to client) [16], [17]. The flow of `SRP-pFed` is given in Figure 2. Our contributions are summarized as follows:

- We propose a Sparsified Random Partial Update framework for PFL, achieving communication-efficient personalized federated training. The update rate of our scheme is coarsely initialized and subsequently refined over the spatial-temporal dimensions instead of empirically set as a fixed constant. To further enhance model personalization, we categorize clients iteratively according to the model weights, and clients in the same group share a common update rate.
- We provide a theoretical analysis of the `SRP-pFed` model training framework, by developing a generic template encompassing training schemes that can be combined with other partial model update approaches–and prove its convergence.
- We compare the performance of our approach to existing methods using five benchmark datasets: CIFAR-10/100 [18], FEMNIST [19], ImageNet [20], and AG News [21], with varying degrees of heterogeneity. The results confirm that `SRP-pFed` consistently outperforms state-of-the-art methods in both accuracy and communication efficiency.

II. Related Work

Challenges of FL: Built upon the distributed system, the performance of FL is constrained by challenges such as data heterogeneity and limited communication [22], [23]. To overcome these hurdles, numerous methods have been proposed [24]: diverse enhanced optimization techniques were explored at the local and global side, such as dynamic regularizer [25], client scheduling, adaptive optimization [26], [27], to address the data heterogeneity; and techniques like pruning [28], as well as the partial model update [29], are adopted to alleviate the communication burdens [24]. Specifically, to explore the performance of FL under realistic data [30], many efforts have been made to realize robust FL training frameworks over noisy, long-tailed, and multi-domain data [31]–[33]. Moreover, techniques like masking [34], dynamic fraction [35], and sparsification [16] have been investigated to improve communication efficiency performance [36].

Personalization in FL: In the presence of data heterogeneity, many approaches have explored personalization methods in FL

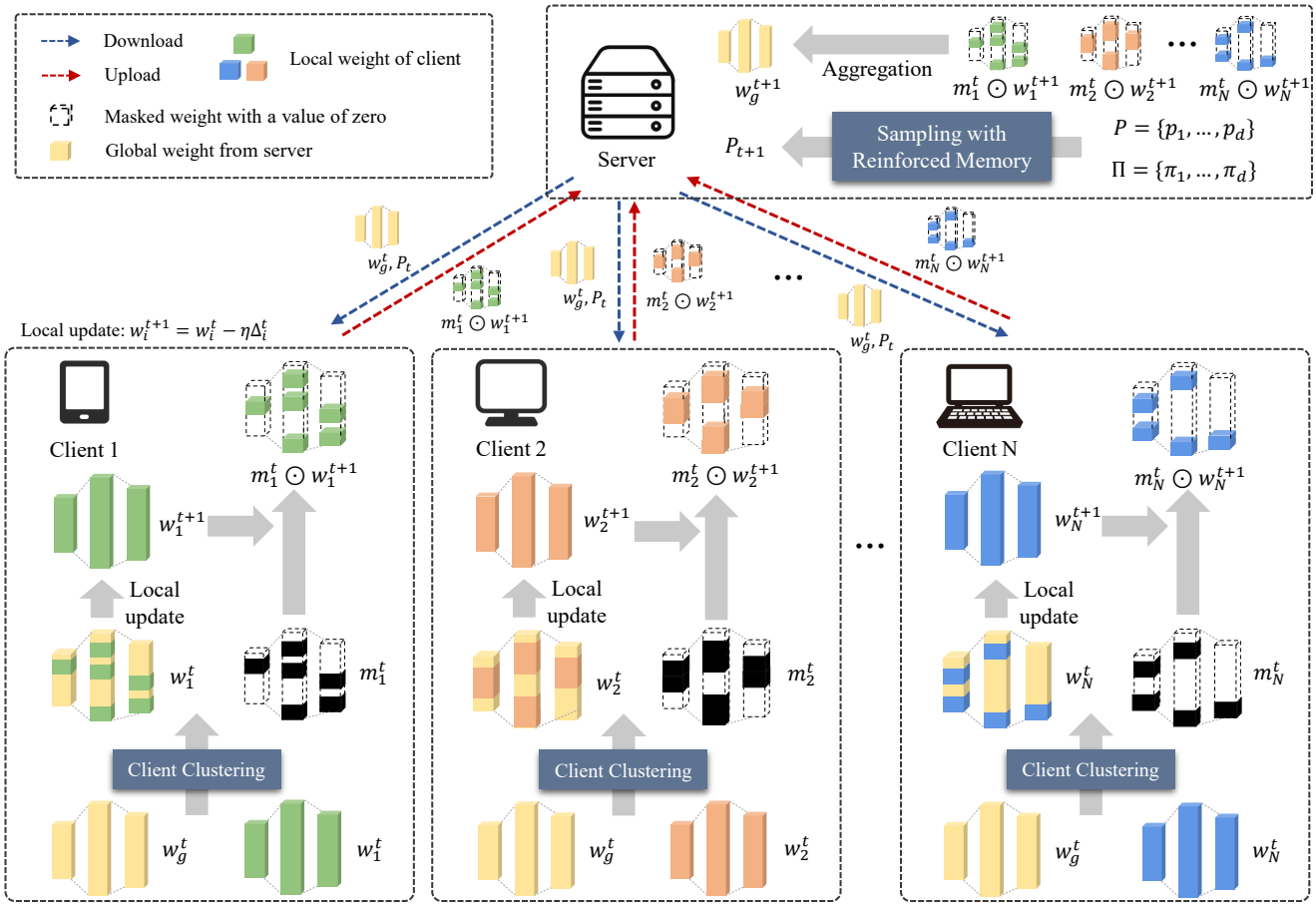

Fig. 2. Overview of SRP-pFed. The server initially samples the update rate set P_t (function SAMPLEUPDATERATE of Algorithm 2) and sends the global model and P_t to each client. Subsequently, each local client runs the client clustering module (function CLIENTCLUSTERING of Algorithm 2) and conducts the local update (function LOCALUPDATE of Algorithm 2). Finally, the updated local model's shared portion is uploaded back to the server.

to deal with the limited generalization capability of a single global model across diverse local clients [4]. In addition to the aforementioned works, multiple PFL works have adopted a multi-task learning framework. For example, Li et al. [37] proposed a two-stage federated optimization framework Ditto to realize robustness and fairness; the clustering is utilized to learn cluster-level personal models [11], [38], [39]. In addition to the aforementioned partial model personalization approaches, diverse federated optimization techniques including the Gaussian process [40], Moreau Envelopes [41], local-global model mixture (i.e., mode interpolation) [10], meta learning [42], and hypernetworks [43], have been explored to enhance the model performance in the context of PFL. Recently, model decoupling approaches have been explored to enhance the personalization performance via model decoupling, in which the system would maintain a global generic model and multiple personal models simultaneously [44]–[46].

Sparsification of DNN: Due to the highly over-parametrized DNNs, training these models in a distributed manner requires intensive computation and large communication overhead. With computation capability having improved significantly more than network bandwidth over the past years [47], communication becomes the bottleneck of distributed learning, particularly at the network edge. To mitigate the amount of transmitted data and expedite training in distributed learning, the sparsification technique, namely pruning, is widely employed. For example, Lottery ticket hypothesis [48], FedDST [16], FedSMP [49] and PerFedMask [8] extract and train sparse **unstructured** sub-networks from the target full network; CD^2-pFed [14] proposes a cyclic distillation-guided **structured** channel decoupling framework; Fan et al. [50] conduct mutual information-based layer-wise pruning. SplitGP [51] adopts the concept of Split Learning, dividing the entire model into client-side and server-side components on a per-layer basis.

The above sparsification approaches reduce the communication overhead and personalize the global model in FL to some extent. Nevertheless, these existing techniques usually preset an update rate empirically and do not consider the changes in the global informativeness demand of the local model during the training process, which is fundamentally different from our SRP-pFed.

III. PROPOSED METHOD

A. System Model

*1) **Personalized Federated Learning**:* Consider an FL system comprised of a server and N clients, in which client i possesses a loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ constructed from

its local dataset $\mathcal{D}_i = \{(\boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R})\}_{i=1}^{n_i}$ with n_i data samples. Each participating entity in this system trains a personalized model by exchanging its model parameters with the server instead of sharing the raw dataset. The overall objective can be formally written as

$$\min_{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N} f(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N) = \sum_{i=1}^{N} \frac{n_i}{n} f_i(\boldsymbol{w}_i), \quad (1)$$

where $f_i(\boldsymbol{w}_i) = \mathbb{E}_{(\boldsymbol{x}_i, y_i) \sim \mathcal{Z}_i}[f_i(\boldsymbol{w}_i; \boldsymbol{x}_i, y_i)]$, $(\boldsymbol{x}_i, y_i) \in D_i, \mathcal{Z}_i$ is the data distribution of client i, $\boldsymbol{w}_i$ indicates client i's model weights after local training, $n_i = |\mathcal{D}_i|$ denotes the number of local data samples at client i, and $n = \sum_{j=1}^{N} |\mathcal{D}_j|$ is the total number of training samples across the clients.

*2) **Partial Model Update**:* We adopt a partial model update method that facilitates model personalization, addresses data heterogeneity, and simultaneously reduces communication overhead. Specifically, every client separates a portion of its local model to share with the server, while the remaining parts are dedicated to personalization. More formally, based on the local model $\boldsymbol{w}_i^t$ (which is trained in the t-th communication round), client i furnishes a mask vector $\boldsymbol{m}_i^t \in \{0,1\}^d$ and constructs the globally shared weights as:

$$\boldsymbol{w}_{s,i}^t = \boldsymbol{m}_i^t \odot \boldsymbol{w}_i^t \quad (2)$$

where $\odot$ denotes the Hadamard product. We define the ratio of the shared weights to the total model weights as the *update rate*:

$$p^t = \frac{\|\boldsymbol{w}_{s,i}^t\|_0}{\|\boldsymbol{w}_i^t\|_0}, \quad (3)$$

where $\|\boldsymbol{w}_{s,i}^t\|_0$ and $\|\boldsymbol{w}_i^t\|_0$ denote the number of shared and personal weights in communication round t, respectively.

In this study, we explore the configuration of the mask $\boldsymbol{m}_i^t$, which is determined based on the magnitude of the absolute values of the weights. Specifically, the weight mask of the local model $\boldsymbol{w}^t$ is constructed by setting the weights with absolute values in the top $100 \times (1 - p^t)\%$ to zero, while the remaining weights are assigned value one. During the model update phase, only the shared model weights are communicated between the clients and the server. Notably, as per the update rate sampling strategy delineated in Section III-B, the critical threshold for determining the zero or one mask in each round is dynamically adjusted. This approach marks a significant departure from the static threshold methods previously proposed [52].

Upon receiving model updates from the selected clients, the server decomposes the model into separate elements and performs aggregation element-wise.[1] Every element is averaged with a weight proportional to the local dataset sizes over the (selected) whole. As such, the global model is updated as follows:

$$\boldsymbol{w}_g^t = \sum_{i \in S_t} \frac{n_i}{\sum_{j \in S_t} n_j} \boldsymbol{m}_i^t \odot \boldsymbol{w}_i^t, \quad (4)$$

where S_t is the set of clients selected for model training in round t, n_i represents the number of local data samples at client i, and $\boldsymbol{w}_i^t = \boldsymbol{w}_i^{t-1} - \eta \sum_{b=1}^{B} \nabla f_i(\boldsymbol{w}_i^{t,b}, z_i^b)$, $z_i^b \subseteq \mathcal{D}_i$, indicates the local model weights of client i.

[1]We can consider different fine-grained model partition strategies (e.g., layer-wise, channel-wise, or element-wise) presented in our algorithm without affecting its generality.

TABLE I
SUMMARY OF IMPORTANT NOTATIONS.

Notation	Meaning (at communication round t)
N	The total number of clients
n	The total number of training samples across the system
n_i	The number of local data samples at client i
$\boldsymbol{w}_g^t$	The global model parameter
$\boldsymbol{w}_i^t$	The model parameter of client i
p_i^t	Model update rate of client i
$\boldsymbol{m}_i^t$	Model mask of client i
S_t	The set of selected clients
f_i	Local loss function of client i
$\mathcal{P}$	Pre-defined candidate set of update rates
$\mathcal{P}_t$	The set of selected update rates
Π	Probability vector of being selected
B	Total number of local data batches
z_i^b	The b^{th} local batch sample chosen from $\mathcal{D}_i$

The server then feeds the global model back to each client for the next round of local training. Consequently, the local model $\boldsymbol{w}_i^t$ at client i is updated as:

$$\boldsymbol{w}_i^t \longleftarrow \boldsymbol{m}_i^t \odot \boldsymbol{w}_g^t + (\boldsymbol{1} - \boldsymbol{m}_i^t) \odot \boldsymbol{w}_i^t, \quad (5)$$

where $\boldsymbol{1} \in \mathbb{R}^d$ is an all-ones vector. For ease of expression, we list the important notations in Table I, and t in each notation denotes the index of the communication round without any confusion.

*3) **Overview of SRP-pFed**:* As discussed in Section I, the amount of global information needed by a client varies during different training phases. Even within the same training phase, the amount of required global information differs from client to client due to data heterogeneity. This variability hinders the conventional fixed update rate from discovering the optimal personalized architecture. Hence, we introduce `SRP-pFed`, a PFL framework that dynamically allocates update rates in the spatio-temporal dimension. This framework comprises two major modules:

- **Update Rates Sampling:** We adjust the update rates in the temporal dimension according to a random walk process [53], [54], renewing the proportion of personal weights and global shared weights in each communication round. Additionally, the module logs the update rate of each selected round and the system's performance to adjust the direction of the next update. This approach enhances system performance by increasing the likelihood of selecting update rates that have proven effective in previous rounds, a concept termed *reinforced memory*. Depending on variations in the distribution of client data, this module may sample one or more update rates.
- **Client Clustering:** Regarding the large variations among clients (e.g., high degree of non-IID or massive client scenarios), we consider sampling multiple update rates per round to better align with each client's requirements. This module divides the clients into distinct groups,

Algorithm 1 An overview of SRP-pFed.

Input: Update rate candidates $\mathcal{P}$, the number of cluster K
Output: Local model $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_N\}$

1: Initialize $\boldsymbol{w}_g^1, \{\boldsymbol{w}_1^1, \boldsymbol{w}_2^1, \ldots, \boldsymbol{w}_N^1\}$
2: **for** each round t from 1 to T **do**
3: Randomly select a subset of clients $\mathcal{S}_t$
4: Update probability vector Π by Equation (7)
5: $\mathcal{P}_t \longleftarrow$ SAMPLEUPDATERATES$(\Pi, \mathcal{P})$
6: **for** each client $i \in \mathcal{S}_t$ in parallel **do**
7: $\boldsymbol{w}_i^t, \boldsymbol{m}_i^t \longleftarrow$ CLIENTCLUSTERING$(\boldsymbol{w}_g^t, \boldsymbol{w}_i^t, \mathcal{P}_t)$
8: $\boldsymbol{w}_i^{t+1} \longleftarrow$ LOCALUPDATE$(\boldsymbol{w}_i^t)$
9: Send partial model $\boldsymbol{m}_i^t \odot \boldsymbol{w}_i^{t+1}$ to the server.
10: **end for**
11: $\boldsymbol{w}_g^{t+1} \longleftarrow \sum_{i \in S_t} \frac{n_i \boldsymbol{m}_i^t \odot \boldsymbol{w}_i^{t+1}}{\sum_{j \in S_t} n_j}$
12: **end for**

where clients within the same group share an identical update rate to minimize the loss function.

Training Process: This part elaborates on the training process (Algorithm 1) of SRP-pFed and the subroutines (Algorithm 2) of it. At the beginning of communication round t, the server executes the sampling module (Line 3 of Algorithm 1) and broadcasts the resulting update rate and global model to each client. Next, the clustering module is executed on the client side. In particular, each client combines local and global models using the received update rate, generating multiple fused models and the corresponding mask (Line 5 of Algorithm 1). Subsequently, the loss of these models on the local data is computed separately, and the model with the smallest loss is selected to execute the local updates (Line 8 of Algorithm 1). Finally, the server gathers the shared components of the client model, filtered through (2), and proceeds with aggregation. As the clients are unburdened from uploading the full model, this process substantially reduces the communication overhead. The above steps are executed iteratively until convergence.

It is worth noting that our design, which involves uploading part of the model but downloading the full model during training, addresses the communication bottleneck caused by the asymmetry in network connection speeds in federated systems. Specifically, uplink (client-to-server) transmission speeds are typically slower than downlink (server-to-client) speeds [16], [17]. For instance, according to the latest 2024 speed test report, the global median uplink speed for mobile connections is 11.33 Mbps, while the median downlink speed is 52.87 Mbps [55]. In contrast, usual FL implementations have symmetric model updates for both directions. Therefore, reducing uplink communication costs while retaining full downlink transmission is a natural strategy for achieving high communication efficiency in resource-constrained edge networks.

B. Update Rate Sampling with Reinforced Memory

In this study, we leverage a random walk model to design the update rate sampling mechanism. The random walk process is a mathematical model that projects the movement

Algorithm 2 Subroutines for SRP-pFed.

function SAMPLEUPDATERATE$(\Pi, \mathcal{P})$
Require $\mathcal{P} = \{p_1, p_2, \ldots, p_c\}$ denotes a pre-defined candidate set of update rates
Require $\Pi = [\pi_1, \pi_2, \ldots, \pi_c]$ represents the normalized probability of each candidate being selected
1: Calculate cumulative probabilities $F_j = \sum_{x=1}^{j} \pi_x$
2: Generate K random numbers $\{U_1, \ldots, U_K\}, U_k \in (0, 1]$
3: **for** U in $\{U_1, \ldots, U_K\}$ **do**
4: Identify the smallest index j such that $F_j \geq U$
5: **end for**
6: Identify the K update rates corresponding to the smallest indices and assemble $\mathcal{P}_t$
7: **return** $\mathcal{P}_t$

function CLIENTCLUSTERING$(\boldsymbol{w}_g^t, \boldsymbol{w}_i^t, \mathcal{P}_t)$
Require $\boldsymbol{w}_g^t$ is latest global model
Require $\boldsymbol{w}_i^t$ is latest local model of client i
Require $\mathcal{P}_t$ is the subset of selected update rates
1: Initialize $loss_{\min} = \infty$
2: **for** p in $\mathcal{P}_t$ **do**
3: Obtain $\boldsymbol{m}$ based on p, where $\|\boldsymbol{m}\|_1 = p \times \|\boldsymbol{w}_i^t\|_0$
4: $\boldsymbol{w} \longleftarrow \boldsymbol{m} \odot \boldsymbol{w}_g^t + (\mathbf{1} - \boldsymbol{m}) \odot \boldsymbol{w}_i^t$
5: **if** $f_i(\boldsymbol{w}) < loss_{\min}$ **then**
6: $loss_{\min} \longleftarrow f_i(\boldsymbol{w})$, $\boldsymbol{w}_{\min} \longleftarrow \boldsymbol{w}, \boldsymbol{m}_{\min} \longleftarrow \boldsymbol{m}$
7: **end if**
8: **end for**
9: **return** $\boldsymbol{w}_{\min}, \boldsymbol{m}_{\min}$

function LOCALUPDATE$(\boldsymbol{w}_i^t)$
Require $\boldsymbol{w}_i^t$ is latest local model of client i
1: $\boldsymbol{w}_i^{t,1} \longleftarrow \boldsymbol{w}_i^t$
2: **for** each local batch b from 1 to B **do**
3: $\boldsymbol{w}_i^{t,b+1} \longleftarrow \boldsymbol{w}_i^{t,b} - \eta \nabla f_i(\boldsymbol{w}_i^{t,b}, z_i^b)$
4: **end for**
5: $\Delta_i^t = \sum_{b=1}^{B} \nabla f_i(\boldsymbol{w}_i^{t,b}, z_i^b)$
6: $\boldsymbol{w}_i^{t+1} \longleftarrow \boldsymbol{w}_i^t - \eta \Delta_i^t$
7: **return** $\boldsymbol{w}_i^{t+1}$

of an object or system from its current position based on a certain probability in discrete time steps (e.g., moments $t = 0, 1, 2, \ldots$). The movement at each step may follow a random pattern and adhere to the laws of a specific probability distribution. Changes in the update rate in the temporal dimension can be conceptualized as the process of altering the object's position. Intuitively, the variation in the update rate should not be haphazard or irregular; instead, it should be a process aimed at enhancing the model's performance. Hence, we incorporate a random walk with a reinforced memory process to characterize the impact of past experiences or states on the current decision after each step. This influence is typically achieved by augmenting the states or decisions associated with prior successes.

To construct a random walk with reinforced memory process, first define the state space representation $\mathcal{P} = \{p_1, p_2, \ldots, p_c\}$ where each p_x represents a potential update rate, c is the total number of candidates of the update rate in the system. At each communication round t, the probability dis-

tribution can be represented by a vector $\Pi = [\pi_1, \pi_2, \ldots, \pi_c]$, where π_x is the probability of p_x being selected. The probability vector is updated based on the historical performance of each state. In particular, let $h^t(x)$ be a time-evolving weight, initialized by setting $h^1(\cdot) = 1$, that captures the historical performance of state p_x up to round t: If p_x is selected, $h(x)$ undergoes a change based on the loss value of the round; otherwise, $h(x)$ remains constant. Formally, this process can be expressed as follows:

$$h^{t+1}(x) = \begin{cases} \lambda h^t(x) + b(t), & \text{if} \quad p_x \in \mathcal{P}_t, \\ \lambda h^t(x), & \text{if} \quad p_x \notin \mathcal{P}_t, \end{cases} \tag{6}$$

where $\mathcal{P}_t \subseteq \mathcal{P}$ denotes the set of update rates selected in communication round t, $b(t) = 1 - \frac{1}{1+e^{-\sum_{i \in S_t} f_i(\boldsymbol{w}_i{}^t)}}$ denotes the weight increment factor that is inversely proportional to the loss in round t, and $\lambda \in (0, 1)$ is a memory decay exponent designed to adapt to the dynamic changes in the system environment. Intuitively, update rates proven effective in the preceding periods should be selected more frequently. Therefore, $h(x)$ is normalized to update the probability π_x as follows

$$\pi_x = \frac{h^t(x)}{\sum_{j=1}^{d} h^t(j)}. \tag{7}$$

Next, update rates $\mathcal{P}_t$ are sampled from the candidate set $\mathcal{P}$ according to the normalized probability Π.

Sampling Procedure: (line 3 of Algorithm 1) Initially, calculate cumulative probabilities $F_j = \sum_{x=1}^{j} \pi_x$, where F_j denotes the sum of normalized probabilities for the first j elements. Next, generate $K < d$ (corresponding to the number of client clustering centers) random numbers from a uniform distribution between 0 and 1, denoted as $U_1, U_2, \ldots, U_K$. For each generated random number U_k, identify the smallest index j such that $F_j \geq U_k$. The update rate p_j is then chosen. This ensures that the selected elements adhere to the original normalized probability distribution. Ultimately, the chosen K update rates constitute the set $\mathcal{P}_t = \{p[1], \ldots, p[K]\} \subseteq \mathcal{P}$.

C. Client Clustering

A common clustering approach involves partitioning the global model $\boldsymbol{w}_g^t$ on the server side according to the sampled update rates, resulting in K sparse models that are then broadcasted to the clients [11]. However, this approach significantly amplifies the communication overhead. In this work, we propose to perform clustering on the client side as shown in Figure 3. The server side only needs to broadcast a global model, and the update rate set $\mathcal{P}_t$ to the client side with little additional communication overhead. The client integrates the global and local models at various proportions based on the received update rates according to Equation (5) and computes the loss for each fused model. The model with the minimum loss is chosen for the local update, and the corresponding mask is retained, as shown below

$$(\boldsymbol{w}_i^t, \boldsymbol{m}_i^t) \longleftarrow \underset{\boldsymbol{w}, \boldsymbol{m}}{\arg\min} \{f_i(\boldsymbol{w}[1]), \ldots, f_i(\boldsymbol{w}[K])\}, \tag{8}$$

where $\boldsymbol{w}[k] = \boldsymbol{m}[k] \odot \boldsymbol{w}_g^t + (1 - \boldsymbol{m}[k]) \odot \boldsymbol{w}_i^t$ denotes the fused model. Subsequently, the mask $\boldsymbol{m}_i^t$ is applied to filter out (by Equation (2)) the shared component of the locally updated model for transmission to the server. This approach only requires uploading a portion of the model, thereby mitigating communication overhead in comparison to traditional federated learning. Client clustering might entail a moderate increase in local computation due to the computation of the loss for K models. However, this is an acceptable trade-off when contrasted with the substantial reduction in communication load.

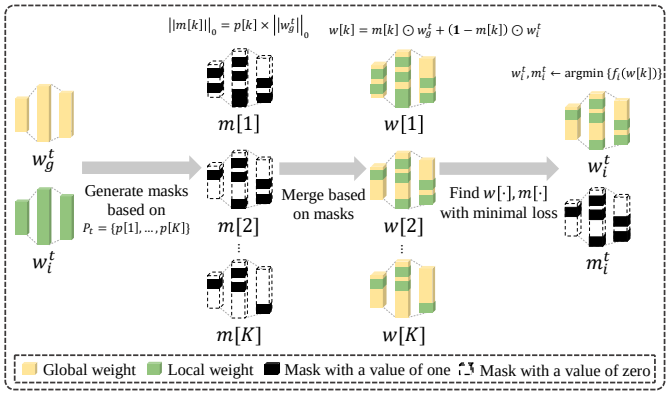

Fig. 3. Client clustering module (function CLIENTCLUSTERING of Algorithm 2). This module merges the global model and local model based on the sampled update rates (Line 4 of CLIENTCLUSTERING), computes the loss for each of the fused models, and outputs the model with the lowest loss along with its corresponding mask (Line 6 of CLIENTCLUSTERING).

D. Adaptation in (Semi)-Asynchronous FL

This paper takes *synchronous* FL as an example to describe the workflow of `SRP-pFed`, which, in fact, can be combined with any PFL approaches to improve the performance of FL systems. The advantage of synchronous FL lies in the consistent updating model, where all clients' updates are aggregated simultaneously, ensuring the model is based on the same global model, achieving more stable and accurate convergence. However, it also leads to long waiting times and high communication overhead, which becomes a critical bottleneck in resource-limited environments. In contrast, asynchronous FL [56] addresses the communication overhead issue but results in some clients updating based on outdated global models, slowing down convergence and affecting model accuracy. Semi-asynchronous FL [57], as a compromise, alleviates the issues of long waiting times or slow convergence. It forces synchronization of outdated local models while performing global updates asynchronously. However, finding the optimal balance between the two is challenging.

In certain scenarios, e.g., devices with varying computational resources, (semi-)asynchronous FL mechanisms may be more advantageous. `SRP-pFed`, as a versatile plug-and-play module, can also perfectly adapt to these scenarios. Specifically, under semi-asynchronous FL, the server aggregates only the updates within a predefined time window and sends the updated global model to each client. By incorporating the `SRP-pFed` module, the system workflow is as follows: (1) The server aggregates the model parameters collected within

the time window, samples the update rates, and returns the update rates and global model to the clients. (2) Regardless of whether the uploaded parameters are adopted, once a client receives the global model and update rates, it sequentially performs clustering and local updates. Asynchronous FL can be viewed as semi-asynchronous FL with an infinitely small time window and can also integrate the `SRP-pFed` module. It can be seen that `SRP-pFed` has broad application scenarios and practical significance. Due to space limitations, we mainly report the performance of this method in synchronous FL scenarios in Section V. We will report the adaptation results in different systems in future work.

IV. Convergence Analysis

This section presents a theoretical analysis of the convergence performance of `SRP-pFed`.

In contrast to conventional FL problems, where learning models $\boldsymbol{w}_i, i \in [N]$ are assumed to be identical for all clients, the heterogeneous data setting results in distinctive learning models across the clients. Following a similar approach as [8], we decompose each learning model into a global representation model and a client-specific model. Given that only the global representations are exchanged with the server, the personalized models are obtained by minimizing the objective function, as Equation (1) shown. Evidently, the loss value of the personalized solution for each client must be smaller than the loss of using one global solution for each client, i.e., $f_i(\boldsymbol{m}_i \odot \boldsymbol{w}_g + (\mathbf{1} - \boldsymbol{m}_i) \odot \boldsymbol{w}_i) < f_i(\boldsymbol{w}_g)$. As long as the global model $\boldsymbol{w}_g$ converges, the personalized model $\boldsymbol{w}_i$ also converges. Therefore, it is sufficient to consider the following objective function

$$f(\boldsymbol{w}_g) = \sum_{i=1}^{N} \frac{n_i}{n} f_i(\boldsymbol{w}_g). \tag{9}$$

Upon the convergence of $\boldsymbol{w}_g$, every client i can obtain its personalized solution $\boldsymbol{w}_i$ by merging the global and local models. Subsequently, we derive the convergence rate to quantify the efficiency of the proposed training algorithm. To facilitate the analysis, we make the following assumptions, which are commonly used and are consistent with numerous theoretical works in FL.

Assumption 1. *The loss functions $f_i(\boldsymbol{w}), i \in [N]$ are L-smooth, i.e., there exists a constant $L > 0$ such that $\|\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{v})\| \leq L\|\boldsymbol{w} - \boldsymbol{v}\|$, $\forall \boldsymbol{w}, \boldsymbol{v} \in \mathbb{R}^d$.*

Assumption 2. *The gradients of local loss are bounded by a constant G, i.e., $\forall \boldsymbol{w} \in \mathbb{R}^d$, $\|\nabla f_i(\boldsymbol{w})\| \leq G, i \in [N]$.*

Assumption 3. *There exists a global bound on the variance of the gradient estimate of each individual client, meaning that: $\frac{1}{N}\sum_{i=1}^{N} \|\nabla f_i(\boldsymbol{w}) - \nabla f(\boldsymbol{w})\|^2 \leq \sigma^2, i \in [N]$.*

Assumption 4. *The gradient noise introduced by the client's local update is bounded, i.e., the unbiased stochastic gradient of client i satisfies: $\mathbb{E}_{z\sim\mathcal{Z}_i}[\|\nabla f_i(\boldsymbol{w}, z) - \nabla f_i(\boldsymbol{w})\|^2] \leq \xi_i^2, i \in [N], z \subseteq \mathcal{D}_i$.*

We first introduce the following lemma, serving as a stepping stone for the main result.

Lemma 1. *The "drift" of the $\boldsymbol{w}_i^{t,b}$ and $\boldsymbol{w}_g^t$ for any $b = 1, \ldots, B$, can be bounded as follows:*

$$\frac{1}{|S_t|}\sum_{i\in S_t} \mathbb{E}[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b}\|^2] \leq 5B\eta^2(\xi_i^2 + 6B\sigma^2) + 30B^2\eta^2G^2. \tag{10}$$

Proof. According to lemma 3, 6, and 7 of [27], we have

$$\begin{aligned}
&\mathbb{E}[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b}\|^2] \\
&= \mathbb{E}[\|\boldsymbol{w}_i^{t,b-1} - \boldsymbol{w}_g^t - \eta\nabla f_i(\boldsymbol{w}_i^{t,b-1}, \boldsymbol{z}_i^{b-1})\|^2] \\
&\leq (1 + \frac{1}{2B-1})\mathbb{E}[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b-1}\|^2] \\
&+ \mathbb{E}[\|\eta(f_i(\boldsymbol{w}_i^{t,b-1}, \boldsymbol{z}_i^{b-1}) - f_i(\boldsymbol{w}_i^{t,b-1}))\|^2] \\
&+ 6B\mathbb{E}[\|\eta(f_i(\boldsymbol{w}_i^{t,b-1}) - \nabla f_i(\boldsymbol{w}_g^t))\|^2] \\
&+ 6B\mathbb{E}[\|\eta(\nabla f_i(\boldsymbol{w}_g^t) - \nabla f(\boldsymbol{w}_g^t))\|^2] + 6B\mathbb{E}[\|\eta\nabla f(\boldsymbol{w}_g^t)\|^2] \\
&\overset{(a)}{\leq} (1 + \frac{1}{2B-1} + 6B\eta^2L^2)\mathbb{E}[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b-1}\|^2] + \eta^2\xi_i^2 \\
&+ 6B\mathbb{E}[\|\eta(\nabla f_i(\boldsymbol{w}_g^t) - \nabla f(\boldsymbol{w}_g^t))\|^2] + 6B\eta^2\mathbb{E}[\|\nabla f(\boldsymbol{w}_g^t)\|^2],
\end{aligned} \tag{11}$$

where (a) follows by using Assumption 4. Averaging over the selected clients, we obtain the following:

$$\begin{aligned}
&\frac{1}{|S_t|}\sum_{i\in S_t} \mathbb{E}[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b}\|^2] \\
&\leq \Big(1 + \frac{1}{2B-1} + 6B\eta^2L^2\Big)\frac{1}{|S_t|}\sum_{i\in S_t} \mathbb{E}\left[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b-1}\|^2\right] \\
&\quad + \eta^2\sum_{i\in S_t}\xi_i^2 + \frac{6B}{|S_t|}\sum_{i\in S_t} \mathbb{E}\left[\|\eta(\nabla f_i(\boldsymbol{w}_g^t) - \nabla f(\boldsymbol{w}_g^t))\|^2\right] \\
&\quad + 6B\eta^2\frac{1}{|S_t|}\sum_{i\in S_t} \mathbb{E}[\|\nabla f(\boldsymbol{w}_g^t)\|^2] \\
&\overset{(a)}{\leq} \Big(1 + \frac{1}{2B-1} + 6B\eta^2L^2\Big)\frac{1}{|S_t|}\sum_{i\in S_t} \mathbb{E}\left[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b-1}\|^2\right] \\
&\quad + \eta^2\Big(\sum_{i\in S_t}\xi_i^2 + 6B\sigma^2\Big) + 6B\eta^2\mathbb{E}\left[\|\nabla f(\boldsymbol{w}_g^t)\|^2\right] \\
&\leq \Big(1 + \frac{1}{B-1}\Big)\frac{1}{|S_t|}\sum_{i\in S_t} \mathbb{E}\left[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b-1}\|^2\right] \\
&\quad + \eta^2\Big(\sum_{i\in S_t}\xi_i^2 + 6B\sigma^2\Big) + 6B\eta^2\mathbb{E}\left[\|\nabla f(\boldsymbol{w}_g^t)\|^2\right],
\end{aligned} \tag{12}$$

where, using Assumption 3, we obtain the inequality (a). Unrolling the recursion, we obtain the following:

$$\begin{aligned}
&\frac{1}{|S_t|}\sum_{i\in S_t} \mathbb{E}\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b}\|^2 \\
&\leq 5B\eta^2\Big(\sum_{i\in S_t}\xi_i^2 + 6B\sigma^2\Big) + 30B^2\eta^2\mathbb{E}\left[\|\nabla f(\boldsymbol{w}_g^t)\|^2\right].
\end{aligned} \tag{13}$$

Then, applying Assumption 2 leads to the conclusion of the Lemma 1. □

When the mask vectors are determined, we define the term $\gamma_i^t = \max_l(\boldsymbol{k}_i^t)_l$, where $\boldsymbol{k}_i^t = \frac{n_i^t\boldsymbol{m}_i^t}{\sum_{j\in S_t} n_j}$. Note that in the partial client participation scenario, we have $\frac{1}{|S_t|} \leq \gamma_i^t \leq 1, i \in S_t$.

Armed with Lemma 1, we obtain the algorithm's convergence rate as follows.

Theorem 1. *Given a pre-defined total number of communication rounds T, by setting $\eta \leq \frac{1}{L|S_t|^2B}$, the* `SRP-pFed` *algorithm converges as:*

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla f(\boldsymbol{w}_g^t)\|^2] \\
\leq&\frac{2\big(f(\boldsymbol{w}_g^1)-f(\boldsymbol{w}_g^{T+1})\big)}{\eta BT}+2\Phi\sum_{i\in S_t}\Big(d_{\boldsymbol{w}}\gamma_i^t-\sum_{l=1}^{d_{\boldsymbol{w}}}(\boldsymbol{k}_i^t)_l\Big) \\
&+\frac{5\eta^3L^2B^2}{2}\Big(\sum_{i\in S_t}\xi_i^2+6B\sigma^2+6BG^2\Big)+\eta LNB\sum_{i\in S_t}\xi_i^2,
\end{aligned} \tag{14}$$

where Φ is a constant satisfying $|\max_l(\nabla f(\boldsymbol{w}_g^t)\odot\nabla f_i(\boldsymbol{w}_i^{t,b}))_l|\leq\Phi$, for all $i\in S_t, b=1,\ldots,B$.

Proof. Since the function f_i is L-smooth, we have

$$\begin{aligned}
\mathbb{E}[f(\boldsymbol{w}_g^{t+1})]\leq&\mathbb{E}[f(\boldsymbol{w}_g^t)]+\mathbb{E}[\langle\nabla f(\boldsymbol{w}_g^t),\boldsymbol{w}_g^{t+1}-\boldsymbol{w}_g^t\rangle] \\
&+\frac{L}{2}\mathbb{E}[\|\boldsymbol{w}_g^{t+1}-\boldsymbol{w}_g^t\|^2].
\end{aligned} \tag{15}$$

We first find an upper bound for $\|\boldsymbol{w}_g^{t+1}-\boldsymbol{w}_g^t\|^2$. Let $\boldsymbol{k}_i^t=\frac{n_i^t\boldsymbol{m}_i^t}{\sum_{j\in S_t}n_j}$ we have

$$\begin{aligned}
&\mathbb{E}[\|\boldsymbol{w}_g^{t+1}-\boldsymbol{w}_g^t\|^2] \\
\overset{(a)}{=}&\eta^2\mathbb{E}[\|\sum_{i\in S_t}\boldsymbol{k}_i^t\odot\sum_{b=1}^{B}\nabla f_i(\boldsymbol{w}_i^{t,b},\boldsymbol{z}_i^b)\|^2] \\
\overset{(b)}{=}&\underbrace{\eta^2\,\mathbb{E}[\|\sum_{i\in S_t}\sum_{b=1}^{B}\boldsymbol{k}_i^t\odot(\nabla f_i(\boldsymbol{w}_i^{t,b},\boldsymbol{z}_i^b)-\nabla f_i(\boldsymbol{w}_i^{t,b}))\|^2]}_{A_1} \\
&+\underbrace{\eta^2\,\|\sum_{i\in S_t}\sum_{b=1}^{B}\boldsymbol{k}_i^t\odot\nabla f_i(\boldsymbol{w}_i^{t,b}))\|^2}_{A_2},
\end{aligned} \tag{16}$$

where equality (a) results from line 8 and 11 of Algorithm 1. Equality (b) is obtained via basic equality $\mathbb{E}\|\boldsymbol{z}\|^2=\mathbb{E}\|\boldsymbol{z}-\mathbb{E}\boldsymbol{z}\|^2+\|\mathbb{E}\boldsymbol{z}\|^2$ for any random vector $\boldsymbol{z}$. By using Assumption 4, we can obtain an upper bound of A_1 as follows:

$$\begin{aligned}
A_1\leq&|S_t|B\sum_{i\in S_t}\sum_{b=1}^{B}\mathbb{E}[\|\boldsymbol{k}_i^t\odot\nabla f_i(\boldsymbol{w}_i^{t,b},\boldsymbol{z}_i^b)-\boldsymbol{k}_i^t\odot\nabla f_i(\boldsymbol{w}_i^{t,b})\|^2] \\
\leq&|S_t|B\sum_{i\in S_t}\sum_{b=1}^{B}\mathbb{E}[\|\nabla f_i(\boldsymbol{w}_i^{t,b},\boldsymbol{z}_i^b)-\nabla f_i(\boldsymbol{w}_i^{t,b})\|^2] \\
\leq&|S_t|B^2\sum_{i\in S_t}\xi_i^2.
\end{aligned} \tag{17}$$

Also, we have A_2 as follows:

$$\begin{aligned}
A_2\leq&|S_t|B\sum_{i\in S_t}\sum_{b=1}^{B}\|\boldsymbol{k}_i^t\odot\nabla f_i(\boldsymbol{w}_i^{t,b}))\|^2 \\
\leq&|S_t|B\sum_{i\in S_t}\sum_{b=1}^{B}\|\nabla f_i(\boldsymbol{w}_i^{t,b}))\|^2.
\end{aligned} \tag{18}$$

Substituting A_1 and A_2 into Equation (16), we have

$$\begin{aligned}
&\mathbb{E}[\|\boldsymbol{w}_g^{t+1}-\boldsymbol{w}_g^t\|^2] \\
\leq&|S_t|B^2\eta^2\sum_{i\in S_t}\xi_i^2+|S_t|B\eta^2\sum_{i\in S_t}\sum_{b=1}^{B}\|\nabla f_i(\boldsymbol{w}_i^{t,b})\|^2.
\end{aligned} \tag{19}$$

Next, we obtain an upper bound for $\mathbb{E}[\langle\nabla f(\boldsymbol{w}_g^t),\boldsymbol{w}_g^{t+1}-\boldsymbol{w}_g^t\rangle]$ as follows:

$$\begin{aligned}
&\mathbb{E}[\langle\nabla f(\boldsymbol{w}_g^t),\boldsymbol{w}_g^{t+1}-\boldsymbol{w}_g^t\rangle] \\
\overset{(a)}{=}&\mathbb{E}[\langle\nabla f(\boldsymbol{w}_g^t),-\eta\sum_{i\in S_t}\boldsymbol{k}_i^t\odot\sum_{b=1}^{B}\nabla f_i(\boldsymbol{w}_i^{t,b},\boldsymbol{z}_i^b)\rangle] \\
\overset{(b)}{=}&-\eta\sum_{i\in S_t}\sum_{b=1}^{B}\mathbb{E}[\langle\nabla f(\boldsymbol{w}_g^t),\boldsymbol{k}_i^t\odot\nabla f_i(\boldsymbol{w}_i^{t,b})\rangle] \\
\overset{(c)}{\leq}&\eta\mathbb{E}[\sum_{i\in S_t}\sum_{b=1}^{B}(-\gamma_i^t)\langle\nabla f(\boldsymbol{w}_g^t),\nabla f_i(\boldsymbol{w}_i^{t,b})\rangle] \\
&+\eta B\Phi\sum_{i\in S_t}(d_{\boldsymbol{w}}\gamma_i^t-\sum_{l=1}^{d_{\boldsymbol{w}}}(\boldsymbol{k}_i^t)_l) \\
\overset{(d)}{\leq}&-\eta\sum_{b=1}^{B}\mathbb{E}[\langle\nabla f(\boldsymbol{w}_g^t),\frac{1}{|S_t|}\sum_{i\in S_t}\nabla f_i(\boldsymbol{w}_i^{t,b})\rangle] \\
&+\eta B\Phi\sum_{i\in S_t}(d_{\boldsymbol{w}}\gamma_i^t-\sum_{l=1}^{d_{\boldsymbol{w}}}(\boldsymbol{k}_i^t)_l),
\end{aligned} \tag{20}$$

where $\Phi=\min_l(\nabla f(\boldsymbol{w}_g^t)\odot\nabla f_i(\boldsymbol{w}_i^{t,b}))_l$, $\gamma_i^t=\max_l(\boldsymbol{k}_i^t)_l$, denote the minimum or maximum element in the vector, respectively. Equality (c) holds by $\mathbb{E}[\nabla f_i(\boldsymbol{w}_i^{t,b},\boldsymbol{z}_i^b)]=\nabla f_i(\boldsymbol{w}_i^{t,b})$. Using Lemma 1 of [8] and $\frac{1}{|S_t|}\leq\gamma_i^t\leq 1$, we obtain the inequality (c) and (d), respectively. Now, our focus is on determining an upper bound for the term to the right of the inequality (d), we represent it as follows:

$$
\begin{aligned}
&- \mathbb{E}[\langle \nabla f(\boldsymbol{w}_g^t), \frac{1}{|S_t|}\sum_{i\in S_t}\nabla f_i(\boldsymbol{w}_i^{t,b})\rangle] \\
=&\frac{1}{2}\mathbb{E}[\|\frac{1}{|S_t|}\sum_{i\in S_t}(\nabla f_i(\boldsymbol{w}_g^t) - \nabla f_i(\boldsymbol{w}_i^{t,b}))\|^2] \\
&- \frac{1}{2}\mathbb{E}[\|\nabla f_i(\boldsymbol{w}_g^t)\|^2] - \frac{1}{2}\mathbb{E}[\|\frac{1}{|S_t|}\sum_{i\in S_t}\nabla f_i(\boldsymbol{w}_i^{t,b})\|^2] \\
\leq&\frac{1}{2}\frac{1}{|S_t|}\sum_{i\in S_t}\mathbb{E}[\|\nabla f_i(\boldsymbol{w}_g^t) - \nabla f_i(\boldsymbol{w}_i^{t,b})\|^2] \\
&- \frac{1}{2}\mathbb{E}[\|\nabla f_i(\boldsymbol{w}_g^t)\|^2] - \frac{1}{2}\mathbb{E}[\|\frac{1}{|S_t|}\sum_{i\in S_t}\nabla f_i(\boldsymbol{w}_i^{t,b})\|^2] \\
\leq&\frac{1}{2}\frac{L^2}{|S_t|}\sum_{i\in S_t}\mathbb{E}[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b}\|^2] \\
&- \frac{1}{2}\mathbb{E}[\|\nabla f_i(\boldsymbol{w}_g^t)\|^2] - \frac{1}{2}\mathbb{E}[\|\frac{1}{|S_t|}\sum_{i\in S_t}\nabla f_i(\boldsymbol{w}_i^{t,b})\|^2]
\end{aligned} \tag{21}
$$

By combining (15), (19) and (21), we have

$$
\begin{aligned}
\mathbb{E}[f(\boldsymbol{w}_g^{t+1})] \leq& \mathbb{E}[f(\boldsymbol{w}_g^t)] + \frac{L}{2}NB^2\eta^2\sum_{i\in S_t}\xi_i^2 \\
&+ \eta B\Phi\sum_{i\in S_t}(d_{\boldsymbol{w}}\gamma_i^t - \sum_{l=1}^{d_{\boldsymbol{w}}}(\boldsymbol{k}_i^t)_l) - \frac{\eta B}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}_g^t)\|^2] \\
&+ \frac{\eta L^2}{2|S_t|}\sum_{b=1}^{B}\sum_{i\in S_t}\mathbb{E}[\|\boldsymbol{w}_g^t - \boldsymbol{w}_i^{t,b}\|^2] \\
&- \frac{\eta}{2}\sum_{b=1}^{B}\sum_{i\in S_t}(\frac{1}{|S_t|} - L\eta|S_t|B)\|\nabla f_i(\boldsymbol{w}_i^{t,b})\|^2.
\end{aligned} \tag{22}
$$

Since $\eta \leq \frac{1}{L|S_t|^2B}$, we have $-\frac{\eta}{2}\sum_{b=1}^{B}\sum_{i\in S_t}(\frac{1}{|S_t|} - L\eta|S_t|B)\|\nabla f_i(\boldsymbol{w}_i^{t,b})\|^2 \leq 0$. Bringing Lemma 1 results into Equation (22), we have

$$
\begin{aligned}
\mathbb{E}[f(\boldsymbol{w}_g^{t+1})] \leq& \mathbb{E}[f(\boldsymbol{w}_g^t)] + \frac{L}{2}NB^2\eta^2\sum_{i\in S_t}\xi_i^2 \\
&+ \eta B\Phi\sum_{i\in S_t}(d_{\boldsymbol{w}}\gamma_i^t - \sum_{l=1}^{d_{\boldsymbol{w}}}(\boldsymbol{k}_i^t)_l) - \frac{\eta B}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}_g^t)\|^2] \\
&+ \frac{\eta^3L^2B^2}{2}(5\sum_{i\in S_t}\xi_i^2 + 30B\sigma^2 + 30BG^2).
\end{aligned} \tag{23}
$$

By rearranging the terms, we obtain

$$
\begin{aligned}
\mathbb{E}[\|\nabla f(\boldsymbol{w}_g^t)\|^2] \leq& \frac{2}{\eta B}(\mathbb{E}[f(\boldsymbol{w}_g^t)] - \mathbb{E}[f(\boldsymbol{w}_g^{t+1})]) \\
&+ \frac{\eta^3L^2B^2}{2}(5\sum_{i\in S_t}\xi_i^2 + 30B\sigma^2 + 30BG^2) \\
&+ \eta LNB\sum_{i\in S_t}\xi_i^2 + 2\Phi\sum_{i\in S_t}(d_{\boldsymbol{w}}\gamma_i^t - \sum_{l=1}^{d_{\boldsymbol{w}}}(\boldsymbol{k}_i^t)_l).
\end{aligned} \tag{24}
$$

Finally, multiply both sides by $\frac{1}{T}$ and sum over $t = 1, \ldots, T$, we have the conclusion of Theorem 1 □

Remark 1. *By employing the mask vectors in FL, the term* $\sum_{i\in S_t}(d_{\boldsymbol{w}}\gamma_i^t - \sum_{l=1}^{d_{\boldsymbol{w}}}(\boldsymbol{k}_i^t)_l)$ *appears on the right side. Since this term does not scale with the number of communication rounds* T*, it is considered a bias term, which remains a residual in the convergence bound. The variability of this term is contingent upon the configurations of the mask structure and the heterogeneity in data among different devices. Given* `SRP-pFed`*'s primary focus on investigating the allocation of update rates, it demonstrates compatibility with a diverse range of mask generation criteria. This variability can be mitigated by adopting specifically tailored mask generation criteria, such as PerFedMask [8].*

V. Experiments and Evaluation

A. Experimental Settings

1) ***Datasets and Models****:* We assess the performance of our `SRP-pFed` across five benchmark datasets in computer vision (CIFAR-10 [18], CIFAR-100 [18], FEMNIST [19], ImageNet [20]), and natural language processing (AG News [21]) domains. Specifically, CIFAR-10 comprises 60,000 32×32 color images categorized into 10 classes; and CIFAR-100 has the same number of image samples as CIFAR-10 but the categories increase to 100, posing a greater challenge for the classification. ImageNet contains 1,200,000 224×224 images, divided into 1000 classes, intensifying the complexity of training classification models. AG News collected 127,600 news texts and divided them into four categories. In contrast to the above datasets, FEMNIST stands out as a dataset purposefully crafted for federated tasks. It encompasses 805,263 samples distributed among 3,550 clients and features 62 labeling categories, eliminating the necessity for additional data partitioning.

To enhance data heterogeneity at various levels, we adopt a *Dirichlet distribution-based* data partitioning method, which has gained prominence in recent literature [25], [58], [59]. Unlike the conventional pathological data distribution [22], this approach provides flexibility in controlling label distribution imbalance, bringing it closer to real-world scenarios. Specifically, *Dirichlet distribution-based* data partitioning regulates the similarity of local data distribution and address class imbalances through a pair of parameters, denoted as (α, ρ). The smaller the values of α and ρ, the more heterogeneous the partitioned data distributes. Due to its inherent design given the distinct challenges presented by various datasets, we conducted our trials utilizing the LeNet-5 architecture [60] for CIFAR-10 and FEMNIST. For CIFAR-100, we opted for the ResNet-34 architecture [61] to address the specific intricacies associated with this dataset. In addition, for the large-scale dataset ImageNet and the NLP Dataset AG News, we chose EfficientNet-B0 [62] and TextFast [63], respectively.

2) ***Baselines****:* We compared the `SRP-pFed` with seven state-of-the-art (SOTA) methods, i.e., FedAvg, LG-Fed, CD^2-pFed, Ditto [37], FedRep [12], PerFedMask [8], and LotteryFL [15]. FedAvg is a classic FL method. In each communication round, clients download the global model from the server and train the model with local data for several epochs on the device. Then, the updated model will be uploaded to the server for aggregation. Ditto learns local models that global regularization encourages to be close together. Unlike

TABLE II
IMPLEMENTATION DETAILS.

Dataset	Learning rate	Batch Size	Local epochs	Fraction	Clients
CIFAR-10	0.01	64	5	0.1	1000/500/100
CIFAR-100	0.01	32	5	0.1	100
FEMNIST	0.01	512	5	0.1	3550
ImageNet	0.1	512	5	0.1	100
AG News	0.01	64	5	0.1	100

FedAvg and Ditto, which exchange full models, the other methods pertain to either structured or unstructured partial model updates.

- **Structured methods:** (i) LG-Fed aggregates only the parameters of the high-dimensional space in the FL process; (ii) CD^2-pFed performs channel-wise assignments for model personalization; (iii) FedRep and FedPerMask learn a shared data representation across clients and unique local head model parameters for each client.
- **Unstructured methods:** LotteryFL learns sparsified subnetworks of the base model by applying the Lottery Ticket Hypothesis, where only the sparsified subnetworks will be communicated between the server and clients. Implementation details of these methods and `SRP-pFed` can be found in the supplementary materials.

*3) **Implementation Details**:* We conducted our trials using PyTorch 1.7.0 in the Python 3.8.13 environment, leveraging an NVIDIA GeForce RTX 3090 (24GB) GPU with CUDA version 12.2. All results represent averages over three simulation runs for each setting. All additional hyperparameters associated with the compared methods were standardized to regular settings, as detailed in Table II. Unless stated otherwise, the number of clients is consistent across all data heterogeneity settings within the same dataset. Concretely, on the CIFAR-100, ImageNet, and AG News, the number of clients remains 100 for both IID and non-IID settings, while on the FEMNIST dataset, the number of clients is 3550. We account for both massive and moderate client scenarios on the CIFAR-10 dataset, resulting in different numbers of clients in settings with varying degrees of data heterogeneity. Specifically, we set $N = 500$ and 100 (moderate) in the $(\alpha, \rho) = (1, 1)$ and $(\alpha, \rho) = (0.1, 1)$ Dirichlet distribution settings, respectively, and $N = 1000$ (massive) in the other settings.

Regarding implementing other approaches, we adhered to the default settings to ensure fairness in the comparison under the same system settings. For instance, in experiments involving the CIFAR-10 dataset, LG-Fed employs LeNet-5 as the backbone, with the weights of the first two layers kept private, while the weights of the last three layers are shared between clients and the server. In contrast, FedRep exclusively shares the weights of the last layer. For experiments on the CIFAR-10/100 and FEMNIST datasets, LotteryFL removes 20% of the L1-norm minimum parameters in each communication round and halts pruning after a cumulative removal of 50% of the parameters.

*4) **Evaluation Metrics**:* We use two primary metrics to assess performance: (i) *Test accuracy* (%). We calculate the weighted local test accuracies, with weights determined by the respective dataset size ratios. (ii) *Communication cost* (Gb). The cumulative communication cost is measured by the total number of bytes transmitted through uplink (clients-to-server) and downlink (server-to-clients) connections. This metric serves as an indicator of communication efficiency performance. In simpler terms, lower communication costs correspond to higher communication efficiency.

B. Performance Comparison

*1) **Accuracy and Communication Efficiency**:* We first compare the test accuracy under fixed communication rounds and communication efficiency to reach a predetermined test accuracy of our `SRP-pFed` alongside seven baseline methods. In this context, we define the fixed communication rounds by referencing the settings used in LotteryFL [15]. Additionally, to ensure that most methods can be achieved (as some methods may perform below FedAvg in certain settings) and to allow direct comparison with results reported in other papers, we set the predefined test accuracy to an integer approximating FedAvg. To ensure a fair and equitable comparison, we have endeavored to standardize all methods concerning the same model and data configurations. However, it is worth noting that some of the baselines lack comprehensive implementation details, and modifying the network architecture in the official implementation code may lead to significant performance variations. An instance is PerFedMask, where the model partition details for parameter sharing of LeNet-5 were not articulated, posing challenges in replicating comparable results on CIFAR-10 and FEMNIST. Consequently, certain baselines were excluded when transitioning to different datasets, ensuring the reliability of the obtained results.

The training performance of `SRP-pFed` and the respective baselines is documented in Table III (for CIFAR-10/100) and Table IV (for FEMNIST). For CIFAR-10/100, the comparison encompasses IID conditions and three distinct levels of non-IID settings, while for FEMNIST, performance evaluations are conducted under both the officially provided IID and non-IID settings. Within these tables, the highest performance for each setting is denoted in bold, with the symbol "$-$" signifying instances where the method fails to attain the designated target accuracy.

The results in Table III show that `SRP-pFed` achieves the highest test accuracy compared to other methods at different levels of data heterogeneity and client size. Specifically, in massive client scenarios ($N = 1000$), `SRP-pFed`, LotteryFL, and FedRep exhibit the highest test accuracies, exceeding 90% in the IID CIFAR-10 configuration. However, upon transitioning to the non-IID CIFAR-10 setup (with $(\alpha, \rho) = (10, 0.7)$), only SRP manages to maintain an accuracy above 80%, while the other two methods experience a significant decline. In this particular scenario, we set the target accuracy at 60%, representing the maximum achievable accuracy for all approaches. Notably, `SRP-pFed` also demonstrates superior communication efficiency.

TABLE III
TEST ACCURACY (%) UNDER FIXED COMMUNICATION ROUNDS R AND THE COMMUNICATION COST (GB) REQUIRED TO REACH A PREDETERMINED ACCURACY ON IID AND NON-IID CIFAR-10/100. THE HIGHEST ACCURACY FOR EACH SETTING IS BOLDFACED.

Dataset	Method	IID		Dirichlet distribution-based non-IID					
				$(\alpha, \rho) = (10, 0.7)$		$(\alpha, \rho) = (1, 1)$		$(\alpha, \rho) = (0.1, 1)$	
		Acc (R=2000)	*Cost* (60%)	*Acc* (R = 2000)	*Cost* (60%)	*Acc* (R = 2000)	*Cost* (45%)	*Acc* (R = 2000)	*Cost* (50%)
CIFAR-10	FedAvg	61.01±2.47	92.96	62.29±0.42	94.62	46.11±1.69	41.67	55.69±0.39	8.49
	LG-Fed	66.96±2.47	7.62	65.65±0.42	7.86	48.82±0.22	4.44	56.97±0.89	0.90
	CD^2-pFed	70.01±0.09	3.69	68.40±0.23	21.69	51.54±0.25	6.42	60.23±0.31	0.43
	Ditto	72.65±0.04	5.23	70.89±0.08	7.13	71.46±1.19	2.62	77.95±0.29	0.62
	LotteryFL	90.64±0.03	3.41	77.67±0.02	6.15	73.76±0.01	1.75	80.09±0.04	0.06
	FedRep	91.03±0.26	4.85	74.05±0.01	5.23	73.90±0.01	2.43	79.51±0.02	0.50
	SRP-pFed	**93.66**±0.79	**2.74**	**83.39**±0.12	**5.71**	**74.36**±0.15	**0.67**	**86.86**±0.09	**0.04**
		Acc (R = 500)	*Cost* (50%)	*Acc* (R = 500)	*Cost* (50%)	*Acc* (R = 500)	*Cost* (50%)	*Acc* (R = 500)	*Cost* (50%)
CIFAR-100	FedAvg	51.03±0.89	683.03	49.27±0.39	–	53.08±0.89	1111.91	53.95±0.25	794.22
	PerFedMask	52.91±0.11	321.12	50.99±0.09	201.31	51.06±0.07	113.24	40.45±0.03	–
	LG-Fed	53.21±0.78	317.69	55.06±0.45	428.88	64.29±0.08	603.61	60.39±0.33	730.68
	Ditto	57.92±0.18	158.84	53.41±0.31	1191.33	62.05±0.01	953.06	61.21±0.05	841.87
	FedRep	51.25±0.34	285.92	55.58±0.25	142.96	60.67±0.13	262.09	63.32±0.19	333.57
	LotteryFL	51.63±0.27	357.47	50.17±0.12	196.97	51.64±0.03	213.93	72.28±0.08	47.95
	SRP-pFed	**66.52**±0.79	**75.55**	**66.55**±0.16	**77.29**	**67.19**±0.16	**80.32**	**77.99**±0.26	**38.26**

TABLE IV
TEST ACCURACY (%) UNDER FIXED COMMUNICATION ROUNDS R AND COST (GB) TO REACH A PREDETERMINED ACCURACY ON FEMNIST.

Method	IID		non-IID	
	Acc (R = 500)	*Cost* (80%)	*Acc* (R = 500)	*Cost* (80%)
FedAvg	80.24±2.66	246.51	78.76±2.66	–
LG-Fed	82.46±0.34	158.76	80.02±1.59	274.46
Ditto	72.85±0.34	–	81.37±1.03	223.78
FedRep	80.43±0.83	150.28	85.30±1.04	54.45
LotteryFL	86.42±1.23	55.21	85.68±3.09	46.55
SRP-pFed	**91.37**±0.89	**43.78**	**89.69**±2.05	**24.36**

TABLE V
TEST ACCURACY (%) UNDER FIXED COMMUNICATION ROUNDS R AND COST (GB) TO REACH A PREDETERMINED ACCURACY ON IMAGENET.

Method	IID		non-IID	
	Acc (R = 100)	*Cost* (40%)	*Acc* (R = 100)	*Cost* (35%)
FedAvg	39.50±0.67	51.63	34.54±0.86	63.54
LG-Fed	39.54±0.75	29.79	35.38±0.57	25.81
LotteryFL	41.89±0.65	23.82	37.39±0.78	20.85
SRP-pFed	**43.52**±0.54	**14.39**	**40.15**±0.69	**13.40**

TABLE VI
TEST ACCURACY (%) UNDER FIXED COMMUNICATION ROUNDS R AND COST (GB) TO REACH A PREDETERMINED ACCURACY ON AG NEWS.

Method	IID		non-IID	
	Acc (R = 2000)	*Cost* (85%)	*Acc* (R = 2000)	*Cost* (80%)
FedAvg	85.34±0.92	1195.23	77.48±1.45	1219.03
LG-Fed	86.14±0.63	976.69	81.24±1.75	855.13
Ditto	91.75±0.75	610.43	89.54±1.03	488.34
FedRep	91.02±0.23	549.39	89.55±0.54	415.09
LotteryFL	93.87±1.10	457.82	92.54±1.25	228.91
SRP-pFed	**95.44**±0.99	**274.69**	**94.54**±1.45	**122.09**

In the case of a higher level of non-IID, we established moderate client scenarios for convergence, specifically $N = 500$ for $(\alpha, \rho) = (1, 1)$ and $N = 100$ for $(\alpha, \rho) = (0.1, 1)$ across all methods. In the $(\alpha, \rho) = (1, 1)$ non-IID setting, SRP-pFed exhibits a slightly higher test accuracy compared to the current optimal method, FedRep. However, the communication cost required to achieve the target accuracy is less than one-third of FedRep's. In the $(\alpha, \rho) = (0.1, 1)$ non-IID setting, representing the highest level of non-IID, SRP-pFed maintains both the highest test accuracy and communication efficiency. These findings are consistent across Tables III, IV, V and VI, reinforcing the superiority and robustness of SRP-pFed's performance across various scenarios.

*2) **Convergence Performance**:* We further examine the convergence rate (illustrated by the test accuracy versus communication rounds) of SRP-pFed against the best-performing full model update method (Ditto), structured (FedRep), and unstructured (LotteryFL) partial model update methods. Figure 4, 5, and 6 display the comparison results over three datasets. These results validate that SRP-pFed significantly improves the convergence speed, whereby compared to the other methods, it attains the highest test accuracy in most settings. Even though, in some cases, Ditto, LotteryFL, and FedRep can achieve test accuracies comparable to SRP-pFed, they converge significantly slower than SRP-pFed. Specifically, all methods achieve a test accuracy exceeding 70% in the non-IID CIFAR-10 (10, 0.7) setting (as shown in Table III). However, LotteryFL requires 984 rounds to reach a 70% test accuracy, whereas SRP-pFed accomplishes the same test accuracy in just 479 rounds, marking a nearly 50% reduction in communication rounds as Figure 4 shows. Not to mention that Ditto and FedRep fail to reach this test accuracy within 1000 rounds. Additionally, as illustrated in Figure 6, the initial convergence rate of FedRep was faster than SRP-pFed, potentially due to sharing the shallow weights responsible for low-level features, which is more conducive

to the performance improvement of initial models in the early stages. Nevertheless, SRP-pFed soon surpasses FedRep after 300 rounds and finally achieves about +2% accuracy than it.

*3) **Discussion**:* Extensive experiments involving diverse non-IID configurations across three datasets substantiate that our proposed SRP-pFed consistently attains state-of-the-art results while incurring the lowest communication cost. Notably, in scenarios characterized by the highest degree of data heterogeneity, SRP-pFed exhibits the most substantial improvement in test accuracy. Regarding algorithmic design, our framework distinguishes itself from other methodologies by incorporating an adaptive update rate allocation mechanism in both spatial (utilizing a client clustering module) and temporal (employing sampling with the reinforced memory module) dimensions. The client clustering module ensures that clients receive update rates tailored to their characteristics, a crucial aspect in scenarios characterized by high data heterogeneity. The sampling with the reinforced memory module guarantees that clients are consistently assigned appropriate update rates even as the local model undergoes continuous updates. It is reasonable to postulate that the observed performance improvement primarily emanates from these aspects. To further investigate the contributions of the aforementioned primary modules and substantiate their necessity, ablation study have been conducted, as elaborated in Section V-C.

C. Ablation Study

TABLE VII
ABLATION STUDY TO EVALUATE THE EFFECTIVENESS OF EACH COMPONENT ON NON-IID CIFAR-10 $(10, 0.7)$.

RD	**RM**	**CL**	*Acc*	*Cost* (60%)	*Round* (60%)
			65.92±0.31	23.85	1340
		✓	68.72±0.45	21.54	1028
✓			72.13±0.32	10.45	655
	✓		72.93±0.21	8.34	571
✓		✓	73.05±0.30	7.84	365
	✓	✓	**73.81±0.43**	**5.94**	**286**

The SRP-pFed is composed of two basic components to realize the partial model update: (1) Update rate sampling with reinforced memory (**RM**) and (2) Client clustering (**CL**). To verify the necessity of reinforced memory during sampling, we introduce the comparison component (3) Sample update rates randomly with the same probability (**RD**) without dynamic adjustment based on historical information.

We compare the performance of different combinations of the above modules to evaluate the effectiveness of each module concerning the best test accuracy over 500 communication rounds, the required communication cost (Gb) as well as the required communication round to reach the target test accuracy. Note that in scenarios where the **RD** and **RM** modules are omitted (first two lines in Table VII), we rely on empirically predefined rules presented in [14] to artificially set the update rate. As illustrated in Table VII, both of our proposed modules contribute to performance improvement. The test accuracy of models incorporating the **RM** module consistently surpasses those with the **RD** module, and this improvement is achieved with reduced communication cost. A similar trend is observed with the **CL** module, where, under identical conditions, superior performance is consistently achieved when employing the module compared to its absence. Notably, the configuration incorporating both **RM** and **CL** (i.e., our proposed SRP-pFed) attains the highest level of performance.

D. Hyperparameters Setting

In this part, we explore the effects of various hyperparameters involved in the experiments. Notably, due to the stochastic nature of operations involved in the model training process, such as model initialization [64], dropout [65], and stochastic gradient descent [66], performance may fluctuate within a certain range. To minimize the impact of randomness, the results presented in this chapter will be averaged over three trials

*1) **Effects of the Number of Update Rate Candidates**:* As the optimal update rate for each client remains uncertain within any specific interval, we refrain from imposing restrictions on the candidate update rate, considering it to be uniformly distributed in the range $(0, 1]$. For the number of candidate update rates, c, we evaluate the system performance of various settings, with c taking values of 5, 10, 15, and 20 (with $K = 2$). The results, illustrated in Figure 7(a), reveal that SRP-pFed is insensitive to changes in c under the IID setting. Conversely, in the three non-IID settings, a notable improvement in test accuracy is observed when c increases from 5 to 10. A larger value of c corresponds to a finer update rate setting. The observed phenomenon indicates that a more refined update rate setting has a positive influence on model performance, especially in non-IID settings. This observation aligns with the explanation for SRP-pFed's ability to maintain high model performance even in non-IID scenarios, as discussed in Section V-B. However, as c further increases to 15 and 20, test accuracy has no significant change. This suggests that the update rate refinement achieved with $c = 10$ fully meets the model's requirements for the update rate, and there is no necessity to further augment the number of candidate update rates. Consequently, we choose $c = 10$ in this study.

*2) **Effects of the** K **Value**:* The variable K denotes the number of clustering centers. A larger K corresponds to a more refined classification of clients and potentially better model performance. Inevitably, this introduces additional computational overhead. Specifically, SRP-pFed requires performing K forward propagation inferences on the local test set, resulting in additional computations quantified as $\text{FLOPs}_{\text{forward}} \times K \times n_{\text{test}}$. The inherent computation of local training comprises the sum of forward propagation and backpropagation operations, multiplied by the number of training samples and epochs, expressed as $(\text{FLOPs}_{\text{forward}} + \text{FLOPs}_{\text{backward}}) \times \text{Epochs} \times n_{\text{train}}$. Backpropagation, which not only calculates the loss but also derives the weight gradients, incurs more significant computational overhead than forward propagation, i.e., $\text{FLOPs}_{\text{forward}} < \text{FLOPs}_{\text{backward}}$. Furthermore, in the setting of this paper, $\text{Epochs} = 5$, $\frac{n_{\text{test}}}{n_{\text{train}}} \leq \frac{1}{5}$. From

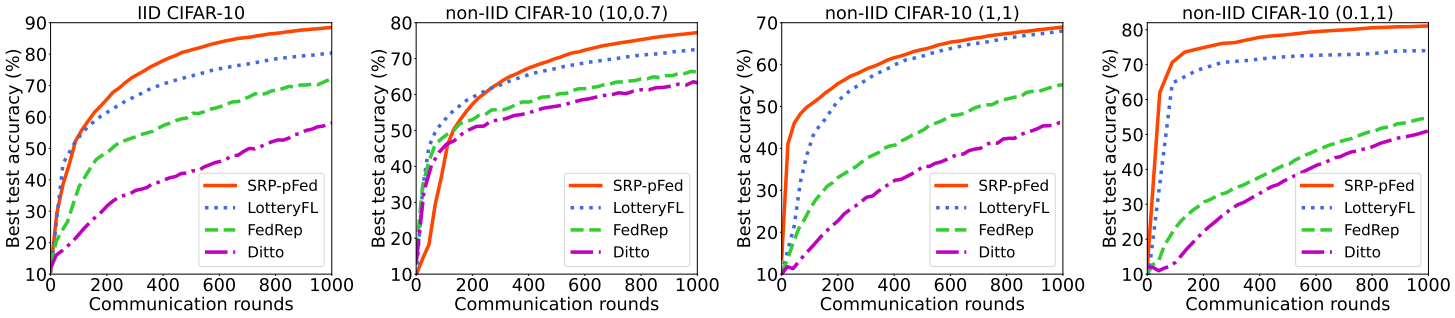

Fig. 4. Test accuracy vs. communication rounds on IID and three non-IID CIFAR-10.

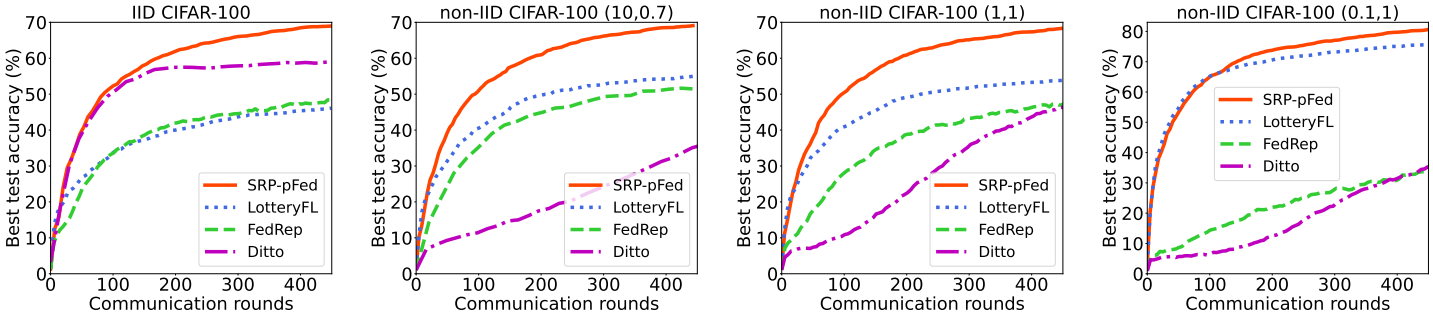

Fig. 5. Test accuracy vs. communication rounds on IID and three non-IID CIFAR-100.

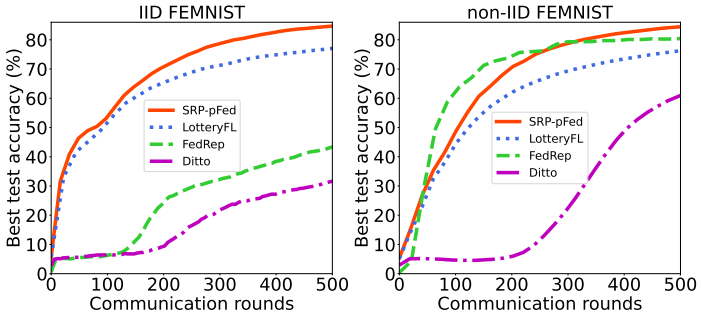

Fig. 6. Test accuracy vs. communication rounds on IID and non-IID FEMNIST.

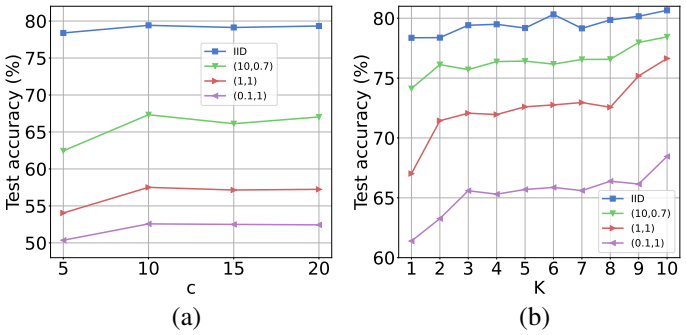

Fig. 7. (a) Sensitivity analysis of the number of update rate candidates c on CIFAR-10, where communication rounds equal to 350. (b) The effect of the number of cluster centers K on CIFAR-10, where communication rounds equal to 200.

these, it can be approximated that the percentage increase in computation due to `SRP-pFed` is

$$\frac{\text{FLOPs}_{\text{forward}} \times K \times n_{\text{test}}}{(\text{FLOPs}_{\text{forward}} + \text{FLOPs}_{\text{backward}}) \times \text{Epochs} \times n_{\text{train}}} \leq \frac{K}{50}.$$

Consequently, it can be observed that 4% increase in computation overhead for each additional clustering center.

To balance model performance and local computation, we conducted trials with a small number of training rounds across different settings on CIFAR-10, exploring various values of K. The outcomes, summarized in Figure 7(b), present a growing trend in test accuracy when `SRP-pFed` adopts more clusters, and the higher the degree of non-IID, the more significant the gain is. While the higher the performance of K the better, the performance gain from the same additional computational overhead varies. The performance increase was most pronounced when K is less than 3 or greater than 8. When $3 \leq K \leq 8$, due to the randomness and insignificant gain, performance shows a fluctuating and slowing upward trend. The reason for this phenomenon is that when the value of K is smaller than the actual number of client clusters (which is unknown), each additional group brings `SRP-pFed`'s clustering closer to the true situation. However, when K exceeds the actual number of clusters, sufficient clustering granularity has already been achieved, and further increasing the number of groups provides only minimal performance gains unless the granularity reaches a certain threshold. In Appendix C, we present a toy example to explain this. In real-world scenarios, due to variations in data distribution, the clustering degree of clients also varies. Determining the optimal value of K

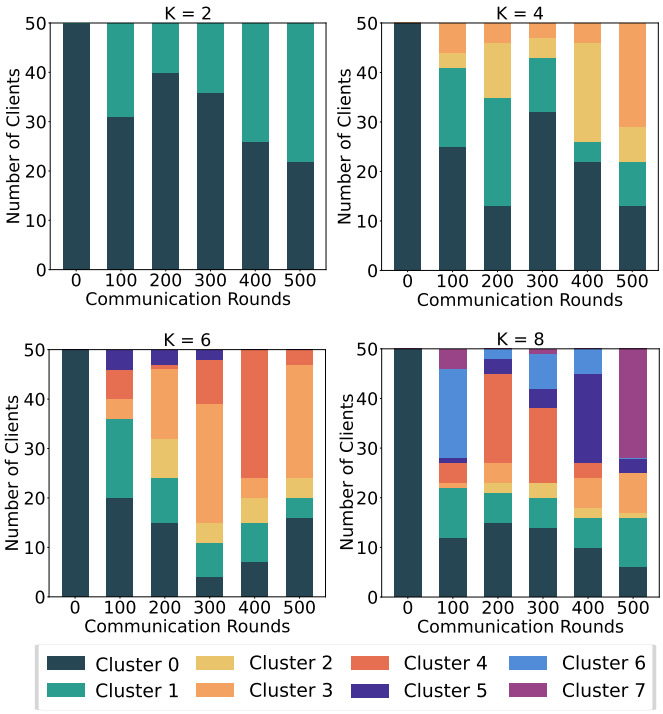

Fig. 8. Visualization of clustering results during training with different K on CIFAR-10 $(1, 1)$.

remains a challenge, and we will explore it in future work.

To explore the cause of the gain slowdown for $3 \leq K \leq 8$, we visualize the clustering during training every 100 communication rounds, as depicted in Figure 8. Initially, at the onset of training (Round = 0), all clients share the same initialization weight, hence the clients are clustered together and not assigned to distinct clustering centers. As the model training progresses, clients are classified into different clusters, reflecting their data heterogeneity. When K is set to 2, clients are more evenly distributed between Clusters 0 and 1. As K increases, clients are more precisely divided to achieve higher accuracy. However, Figure 8 also reveals that some clusters contain very few clients, providing limited performance gains but incurring additional local computation costs. For instance, with $K = 6$ at communication round 500, more than 80% of clients are classified into Clusters 0 and 3, while the remaining (less than 20%) are scattered across the other three clusters. Roughly categorizing this smaller portion into Clusters 0 and 3 has a negligible impact on performance, reducing computational overhead, as illustrated in Figure 7. A similar situation exists for $K = 4$ and $K = 8$. Therefore, considering the trade-off between performance and computational effort, we choose $K = 2$, which yields significant performance gains with minimal additional computation overhead (4%).

E. Limitation and Future Work

This part reports the convergence results of `SRP-pFed` and demonstrates superior performance over existing baselines in synchronous federated learning scenario. However, its adaptability in certain specific scenarios (such as dynamic user numbers and asynchronous FL mechanisms) has yet to be confirmed. In Section III, we briefly discussed `SRP-pFed`'s adaptability in (semi-)asynchronous FL mechanisms. We also provided preliminary results of SRP-pFed under the dynamic client number setting in the Appendix B. In future work, we will further validate the generality and effectiveness of the proposed method in various scenarios through both theoretical analysis and experiments.

VI. Conclusion

We proposed `SRP-pFed`, a PFL framework capable of personalizing FL models while concurrently reducing communication costs. Under the `SRP-pFed`, each client divides its local model into the personal and shared parts, where only the shared part is exchanged with the server. The size of the personal part in a local model is determined by the update rate, which is coarsely initialized and subsequently refined over time. Our experimental results amply demonstrated the effectiveness of the `SRP-pFed` under different data heterogeneity settings, in both massive and moderate client scenarios. Moreover, the developed framework can be extended to other architecture-based PFL approaches.

The current experiment assesses the `SRP-pFed` with a fixed number of clients. However, client participation may not be consistent during training in practical applications. Exploring the feasibility of the proposed system performance under conditions where the number of clients changes dynamically [45], [67], [68] would be insightful. We will explore this aspect in our future work.

References

[1] A. Pothitos, "Iot and wearables: Fitness tracking," 2017. [Online]. Available: http://www.mobileindustryreview.com/2017/03/iot-wearables-fitness-tracking.html.

[2] P. Goldstein, "Smart cities gain efficiencies from iot traffic sensors and data," 2018. [Online]. Available: https://statetechmagazine.com/article/2018/12/

[3] A. Weinreic, "The future of the smart home: Smart homes and iot: A century in the making," 2018. [Online]. Available: https://statetechmagazine.com/article/2018/12/smart-cities-gain-efficiencies-iot-traffic-sensors-and-data-perfcon.

[4] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 34, pp. 9587–9603, 2022.

[5] Y. Xiong, R. Wang, M. Cheng, F. Yu, and C.-J. Hsieh, "Feddm: Iterative distribution matching for communication-efficient federated learning," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 16 323–16 332.

[6] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proc. Nat. Acad. Sci.*, vol. 118, no. 17, p. e2024789118, 2021.

[7] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin, "Rethinking architecture design for tackling data heterogeneity in federated learning," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 10 061–10 071.

[8] M. Setayesh, X. Li, and V. W. Wong, "Perfedmask: Personalized federated learning with optimized masking vectors," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[9] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[10] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," *arXiv preprint arXiv:2002.05516*, 2020.

[11] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 19 586–19 597, 2020.

[12] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *Int. Conf. Mach. Learn. (ICML)*. PMLR, 2021, pp. 2089–2099.

[13] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," in *Intl. Workshop Federated Learn. Data Privacy@NeurIPS*, 2019. [Online]. Available: arXivpreprintarXiv:2001.01523

[14] Y. Shen, Y. Zhou, and L. Yu, "Cd2-pfed: Cyclic distillation-guided channel decoupling for model personalization in federated learning," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 10 041–10 050.

[15] A. Li, J. Sun, B. Wang, L. Duan, S. Li, Y. Chen, and H. Li, "Lotteryfl: Empower edge intelligence with personalized and communication-efficient federated learning," in *ACM/IEEE Symp. Edge Comput*, 2021, pp. 68–79.

[16] S. Bibikar, H. Vikalo, Z. Wang, and X. Chen, "Federated dynamic sparse training: Computing less, communicating less, yet learning better," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 6, 2022, pp. 6080–6088.

[17] J. Konecnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *NIPS Workshop Private Multi-Party ML*, 2016. [Online]. Available: https://arxiv.org/pdf/1610.05492.

[18] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[19] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," in *Workshop Federated Learn. Data Privacy Confid.*, 2019.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis. (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[21] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

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

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

[24] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data *et al.*, "A field guide to federated optimization," *arXiv preprint arXiv:2107.06917*, 2021.

[25] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[26] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," in *Int. Workshop Fed. Learn. Data Priv. NeurIPS*, 2019. [Online]. Available: arXivpreprintarXiv:1909.06335

[27] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[28] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Trans. Neural Netw. Learn, Syst.*, vol. 34, no. 12, pp. 10 374–10 386, 2022.

[29] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn, Syst.*, vol. 31, no. 10, pp. 4229–4238, 2019.

[30] T.-M. H. Hsu, H. Qi, and M. Brown, "Federated Visual Classification with Real-World Data Distribution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, August 2020.

[31] X. Fang, M. Ye, and X. Yang, "Robust heterogeneous federated learning under data corruption," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5020–5030.

[32] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du, "Rethinking federated learning with domain shift: A prototype view," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 16 312–16 322.

[33] Y. Lu, P. Qian, G. Huang, and H. Wang, "Personalized federated learning on long-tailed data via adversarial feature augmentation," in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2023.

[34] S. Ji, W. Jiang, A. Walid, and X. Li, "Dynamic sampling and selective masking for communication-efficient federated learning," *IEEE Intell. Syst.*, vol. 37, pp. 27–34, 2021.

[35] Z. Chen, K. F. E. Chong, and T. Q. Quek, "Dynamic attention-based communication-efficient federated learning," in *Int. Workshop Fed. Transf. Learn. Data Spars. Confidential. IJCAI*, 2021. [Online]. Available: arXivpreprintarXiv:2108.05765

[36] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[37] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *Int. Conf. Mach. Learn. (ICML)*. PMLR, 2021, pp. 6357–6368.

[38] X. Tang, S. Guo, and J. Guo, "Personalized federated learning with clustered generalization," *arXiv preprint arXiv:2106.13044*, 2021.

[39] Z. Li, Z. Chen, X. Wei, S. Gao, C. Ren, and T. Q. Quek, "Hpfl-cn: Communication-efficient hierarchical personalized federated edge learning via complex network feature clustering," in *IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, 2022, pp. 325–333.

[40] I. Achituve, A. Shamsian, A. Navon, G. Chechik, and E. Fetaya, "Personalized federated learning with gaussian processes," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 8392–8406, 2021.

[41] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 21 394–21 405, 2020.

[42] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," *arXiv preprint arXiv:1909.12488*, 2019.

[43] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik, "Personalized federated learning using hypernetworks," in *Int. Conf. Mach. Learn. (ICML)*. PMLR, 2021, pp. 9489–9502.

[44] H.-Y. Chen and W.-L. Chao, "On bridging generic and personalized federated learning for image classification," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[45] Z. Chen, H. Yang, T. Quek, and K. F. E. Chong, "Spectral co-distillation for personalized federated learning," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 8757–8773, 2023.

[46] Z. Xiao, Z. Chen, L. Liu, Y. Feng, J. Wu, W. Liu, J. T. Zhou, H. H. Yang, and Z. Liu, "Fedloge: Joint local and generic federated learning under long-tailed data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[47] L. Luo, J. Nelson, L. Ceze, A. Phanishayee, and A. Krishnamurthy, "Parameter hub: a rack-scale parameter server for distributed deep neural network training," in *Proc. ACM Symp. Cloud Comput.*, 2018, pp. 41–54.

[48] A. Morcos, H. Yu, M. Paganini, and Y. Tian, "One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[49] R. Hu, Y. Guo, and Y. Gong, "Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy," *IEEE Trans. Mobile Comput.*, vol. 23, pp. 8242–8255, 2023.

[50] C. Fan, J. Li, T. Zhang, X. Ao, F. Wu, Y. Meng, and X. Sun, "Layer-wise model pruning based on mutual information," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2021, pp. 3079–3090.

[51] D.-J. Han, D.-Y. Kim, M. Choi, D. Nickel, J. Moon, M. Chiang, and C. G. Brinton, "Federated split learning with joint personalization-generalization for inference-stage optimization in wireless edge networks," *IEEE Trans. Mobile Comput.*, vol. 23, pp. 7048–7065, 2023.

[52] A. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2017, pp. 440–445.

[53] Z.-J. Tan, X.-W. Zou, S.-Y. Huang, W. Zhang, and Z.-Z. Jin, "Random walk with memory enhancement and decay," *Phys. Rev. E*, vol. 65, no. 4, p. 041101, 2002.

[54] E. Baur, "On a class of random walks with reinforced memory," *J. Stat. Phys.*, vol. 181, no. 3, pp. 772–802, 2020.

[55] Speedtest, "Internet speed around the world – speedtest global index speeds," 2024. [Online]. Available: https://www.speedtest.net/global-index.

[56] Z. Wang, Z. Zhang, Y. Tian, Q. Yang, H. Shan, W. Wang, and T. Q. Quek, "Asynchronous federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 6961–6978, 2022.

[57] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, and H. Huang, "Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3654–3672, 2021.

[58] Y. Wu, S. Zhang, W. Yu, Y. Liu, Q. Gu, D. Zhou, H. Chen, and W. Cheng, "Personalized federated learning under mixture of distributions," in *Int. Conf. Mach. Learn. (ICML)*. PMLR, 2023, pp. 37 860–37 879.
[59] O. Marfoq, G. Neglia, A. Bellet, L. Kameni, and R. Vidal, "Federated multi-task learning under a mixture of distributions," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 15 434–15 447, 2021.
[60] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
[62] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Int. Conf. Mach. Learn. (ICML)*. PMLR, 2019, pp. 6105–6114.
[63] A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist.*, vol. 2, 2017, pp. 427–431.
[64] D. Arpit, V. Campos, and Y. Bengio, "How to initialize your network? robust initialization for weightnorm & resnets," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
[65] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[66] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, pp. 400–407, 1951.
[67] J. Luo, M. Mendieta, C. Chen, and S. Wu, "Pgfed: Personalize each client's global objective for federated learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 3946–3956.
[68] S. Wang and M. Ji, "A unified analysis of federated learning with arbitrary client participation," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 19 124–19 137, 2022.

Xinyi Hu (S'18–M'23) received the B.E. degree in Electronic Information Engineering from Xidian University, Xi'an, China, in 2018. She received her Ph.D. degree in Electronic Science and Technology from Zhejiang University, Hangzhou, China, in 2023. Currently, she is a Postdoctoral Research Fellow at Zhejiang University/University of Illinois at Urbana-Champaign Institute (ZJU-UIUC Institute), Zhejiang University, Haining, China. Her research mainly focuses on federated learning, machine learning, and model compression.

Zihan Chen (S'18–M'22) received the B.Eng. degree in Communication Engineering from the Yingcai Honors College at the University of Electronic Science and Technology of China (UESTC) in 2018. He received his Ph.D. degree from the Singapore University of Technology and Design (SUTD)-National University of Singapore(NUS) Joint Ph.D. Program in 2022. Currently, he is a Postdoctoral Research Fellow at SUTD. His research mainly focuses on network intelligence, machine learning, and semantic communication.

Chenyuan Feng (S'16-M'21) received the B.E. degree in electrical and electronics engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2016, and the Ph.D. degree in information system technology and design from Singapore University of Technology and Design (SUTD), Singapore, in 2021, respectively. Currently she is a research fellow at Eurecom, France. Her research interests include edge intelligence, multimedia intelligence, as well as AI for network and communication.

Dr. Feng is also a receipt of Marie Skłodowska-Curie global fellowship (EU Talent Program) and the 2021 IEEE ComComAp Best Paper Award. She was invited to deliver several tutorials and invited talk at International conferences in the area of machine learning for communication, such as IEEE PIMRC'2024, IEEE ICCT'2022 and IEEE ICCT'2024. She also serves as an Editor for the IEEE INTERNET OF THINGS JOURNAL and the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY.

Geyong Min (S'95-M'03) is the Chair Professor and Director of High Performance Computing and Networking (HPCN) Research Group at the University of Exeter, UK. He received the Ph.D. degree in Computing Science from the University of Glasgow, UK, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. He joined the University of Bradford as a Lecturer in 2002, became a Senior Lecturer in 2005 and a Reader in 2007, and was promoted to a Professor in Computer Science in 2012. His main research interests include Next-Generation Internet, Wireless Networks, Mobile Computing, Cloud Computing, Big Data, Multimedia Systems, Information Security, System Modeling and Performance Optimization.

Dr. Min has produced more than 200 research publications including 2 edited books, 18 book chapters, 5 conference proceedings, and 100 papers in the leading international journals including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE NETWORK, IEEE INTELLIGENT SYSTEMS, IEEE INTERNET COMPUTING, AND ACM TRANSACTIONS ON EMBEDDED COMPUTING SYSTEMS, as well as 100 papers at the reputable international conferences, such as SIGCOMM-IMC, ICDCS, IPDPS, GLOBECOM, and ICC.

Dr. Min was the recipient of five Best Paper Awards from IEEE GREENCOM'2013, TRUSTCOM'2010, CSE'2009, ICAC'2008, and AINA'2007 CONFERENCES, respectively. He was invited to deliver 7 keynote speeches and 2 invited talks at International conferences in the area of High Performance Computing and Networking.

Tony Q. S. Quek (S'98-M'08-SM'12-F'18) received the B.E. and M.E. degrees in electrical and electronics engineering from the Tokyo Institute of Technology in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology in 2008. Currently, he is the Cheng Tsang Man Chair Professor with Singapore University of Technology and Design (SUTD) and ST Engineering Distinguished Professor. He also serves as the Director of the Future Communications R&D Programme, the Head of ISTD Pillar, and the Deputy Director of the SUTD-ZJU IDEA. His current research topics include wireless communications and networking, network intelligence, non-terrestrial networks, open radio access network, and 6G.

Dr. Quek has been actively involved in organizing and chairing sessions, and has served as a member of the Technical Program Committee as well as symposium chairs in a number of international conferences. He is currently serving as an Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Dr. Quek was honored with the 2008 Philip Yeo Prize for Outstanding Achievement in Research, the 2012 IEEE William R. Bennett Prize, the 2015 SUTD Outstanding Education Awards – Excellence in Research, the 2016 IEEE Signal Processing Society Young Author Best Paper Award, the 2017 CTTC Early Achievement Award, the 2017 IEEE ComSoc AP Outstanding Paper Award, the 2020 IEEE Communications Society Young Author Best Paper Award, the 2020 IEEE Stephen O. Rice Prize, the 2020 Nokia Visiting Professor, and the 2022 IEEE Signal Processing Society Best Paper Award. He is a Fellow of IEEE and a Fellow of the Academy of Engineering Singapore.

Howard H. Yang (S'13–M'17) received the B.E. degree in Communication Engineering from Harbin Institute of Technology (HIT), China, in 2012, and the M.Sc. degree in Electronic Engineering from Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2013. He earned the Ph.D. degree in Electrical Engineering from Singapore University of Technology and Design (SUTD), Singapore, in 2017. He was a Postdoctoral Research Fellow at SUTD from 2017 to 2020, a Visiting Postdoc Researcher at Princeton University from 2018 to 2019, and a Visiting Student at the University of Texas at Austin from 2015 to 2016. Currently, he is an assistant professor at the Zhejiang University/University of Illinois at Urbana-Champaign Institute (ZJU-UIUC Institute), Zhejiang University, Haining, China. He is also an adjunct assistant professor with the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign, IL, USA

Dr. Yang's research interests cover various aspects of wireless communications, networking, and signal processing, currently focusing on the modeling of modern wireless networks, high dimensional statistics, graph signal processing, and machine learning. He serves as an editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He received the IEEE ComSoc Asia Pacific Oustanding Young Researcher Award in 2023, the IEEE Signal Processing Society Best Paper Award in 2022, the IEEE WCSP 10-Year Anniversary Excellent Paper Award in 2019, and the IEEE WCSP Best Paper Award in 2014.