

Learnable Sparse Customization in Heterogeneous Edge Computing

Jingjing Xue^{1,2}, Sheng Sun¹, Min Liu^{1,2,5,*}, Yuwei Wang¹, Zhuotao Liu^{3,5}, Jingyuan Wang⁴

¹Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

²University of Chinese Academy of Sciences, Beijing, China

³Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China

⁴School of Computer Science and Engineering, Beihang University, Beijing, China

⁵Zhongguancun Laboratory, Beijing, China

Email: {xuejingjing20g, sunsheng, liumin, ywwang}@ict.ac.cn, zhuotao.liu@tsinghua.edu.cn, jywang@buaa.edu.cn

Abstract—To effectively manage and utilize massive distributed data at the network edge, Federated Learning (FL) has emerged as a promising edge computing paradigm across data silos. However, FL still faces two challenges: system heterogeneity (*i.e.*, the diversity of hardware resources across edge devices) and statistical heterogeneity (*i.e.*, non-IID data). Although sparsification can extract diverse submodels for diverse clients, most sparse FL works either simply assign submodels with artificially-given rigid rules or prune partial parameters using heuristic strategies, resulting in inflexible sparsification and poor performance. In this work, we propose Learnable Personalized Sparsification for heterogeneous Federated learning (FedLPS), which achieves the learnable customization of heterogeneous sparse models with importance-associated patterns and adaptive ratios to simultaneously tackle system and statistical heterogeneity. Specifically, FedLPS learns the importance of model units on local data representation and further derives an importance-based sparse pattern with minimal heuristics to accurately extract personalized data features in non-IID settings. Furthermore, Prompt Upper Confidence Bound Variance (P-UCBV) is designed to adaptively determine sparse ratios by learning the superimposed effect of diverse device capabilities and non-IID data, aiming at resource self-adaptation with promising accuracy. Extensive experiments show that FedLPS outperforms status quo approaches in accuracy and training costs, which improves accuracy by 1.28%-59.34% while reducing running time by more than 68.80%.

Index Terms—Edge Computing, Federated Learning, Model Sparsification, System and Data Heterogeneity

I. INTRODUCTION

The proliferation of mobile and IoT devices drives the significant growth of data generated at the network edge [1, 2], which, coupled with the demand for real-time data processing and privacy protection, has fueled the rise of edge data management and utilization [3–5]. Traditional centralized methods require edge devices to upload such a huge amount of data for centralized processing, which tends to exhaust network capacity and bring unacceptable transferring latency [6, 7]. Besides, raw data uploading takes the risk of user privacy leakage and unauthorized access [8, 9]. Faced with these limitations, the landscape of data management has significantly changed, giving rise to embracing Mobile Edge Computing (MEC) in distributed data management across edge users [10]. MEC pushes data storage and model computing to network edges at

the source instead of raw data transmission, enabling real-time processing and safeguarding privacy. As a distributed edge computing paradigm, FL [11–13] allows devices to locally process data and jointly compute a shared model without raw data sharing, which has become a promising solution for privacy-preserving edge data management and utilization [14, 15]. In FL, edge devices (*i.e.*, clients) locally update models on their own data and periodically upload local updates to the server for aggregating into a global model.

Despite the benefits of data localization and privacy protection, FL still faces two key challenges: (1) **System heterogeneity** highlights that different edge devices possess diverse resource configurations, which limits clients to train models matching their capabilities, resulting in severe performance gaps [16, 17]. (2) **Statistical heterogeneity** focuses on non-IID data among clients [18, 19], which leads to apparent inconsistencies between local updates [20], further degrading model generalization [21, 22]. The simultaneous existence of such dual heterogeneity brings additive bottlenecks in processing efficiency and performance, hindering the deployment of FL in practical Edge Data Management (EDM) and MEC scenarios.

Prior works prioritize clients with powerful capabilities [23] and tolerate stale updates [24], which introduces training bias due to poor fairness and sacrifices accuracy considering global model drift. While some studies [25, 26] limit the inconsistencies between local updates and global models to ensure convergence on non-IID data, which still suffer from heavy burden. The primary drawback of both directions is the identical model architecture across all clients, which incurs stragglers slowing down the FL process and brings inference accuracy gaps. Therefore, different local models are advocated to match heterogeneous settings of edge devices [27].

Sparsification is a promising solution for heterogeneous model extraction, which can prune different parameters from the global model to build diverse submodels. A sparse model involves two determining factors: (1) **Sparse ratio** indicates how many parameters are retained after sparsification. (2) **Sparse pattern** points out which parameters are removed, which determines the submodel structure. It has been verified that changes in both sparse ratio and pattern yield notable differences in resource costs and model accuracy [28–30].

* Min Liu is the Corresponding author.

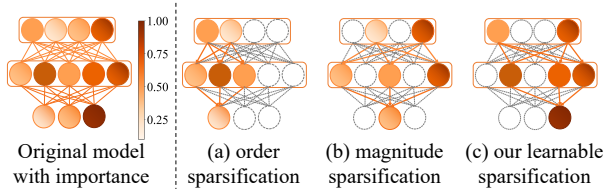


Fig. 1: Different pattern strategies. The padding represents importance scores and no padding indicates the unit is sparsified.

Several pioneering studies [30, 31] introduce sparsification into FL to assign distinct models for clients, where rigid rules of sparse ratio are artificially set based on device capabilities. However, sparse ratios are not only restricted by diverse capabilities, but also closely correlated with model accuracy [28, 32]. These rigid rules cannot flexibly model the interaction between sparse ratio, capability, and accuracy, failing to trade off resource self-adaption and accuracy guarantee over non-IID data. To fit non-IID data, recent works [33–35] explore the personalization of sparse models. They heuristically specify sparse patterns including random [36], ordered [37], and magnitude-based [34, 38] sparsify, ignoring the precise measure of model unit importance on local data. As shown in Figure 1, we use the depth of padding to indicate model unit importance, and these heuristic strategies prune some significant units. The resulting sparse model loses the representation abilities of the corresponding important parameters and cannot accurately extract local data features, resulting in performance degradation. Overall, artificially-given rigid ratio rules and heuristic pattern strategies cannot effectively accommodate complicated dual heterogeneity in real EDM scenarios.

In light of the above observation, we propose Learnable Personalized Sparsification for heterogeneous Federated learning (FedLPS), which enables learnable customization of sparse patterns and sparse ratios to tailor a capability-affordable and accuracy-remarkable submodel for each client. Specifically, FedLPS links the model unit importance and local data in local loss. With the importance-associated loss, the client accurately learns importance indicators based on local data by the back propagation to derive importance-based sparse patterns with minimal heuristics, achieving personalized sparse training. Furthermore, we design P-UCBV with accuracy-dominated arm elimination to learn the correlation among sparse ratio, capability, and accuracy. Based on the superposition effect of diverse capabilities and non-IID data, we adaptively determine sparse ratios for clients to flexibly accommodate all possible cases. In this way, FedLPS allows each client to customize a sparse model with the learnable pattern and adaptive ratio to accurately process personalized data and match diverse capabilities, boosting model performance and computing efficiency.

Our main contributions can be summarized as follows:

- FedLPS takes the first step to customize a data-driven and resource-adapted sparsification in learnable ways for each client so as to accelerate the training process while enhancing inference accuracy in complicated non-IID and system-heterogeneous MEC scenarios.

- For precise and efficient edge data processing, unit-wise importance indicators are optimized on local data to facilitate learnable pattern personalization, and P-UCBV learns additive feedback of time costs and accuracy under diverse resource and non-IID data to adaptively determine computation-efficient and accuracy-guaranteed ratios.
- Theoretically, we provide the upper bound of the gap between global and local parameters in heterogeneous settings and prove the convergence of FedLPS under SGD optimization with the constraints of learning rates.
- We conduct experiments on 5 classic datasets with various models. The results show that FedLPS provides 1.28%-59.34% accuracy gains while achieving up to 60% reduction in computation costs and more than 68.80% time acceleration compared to baselines.

II. PRELIMINARIES AND PROBLEM FORMULATION

Suppose K edge devices with data $\{D_1, \dots, D_K\}$ participate in edge computing. These distributed data are essentially non-IID and devices generally hold heterogeneous resource configurations in real MEC scenarios. Let z_k represent the computation capability of client k .

Personalization Target. To fit non-IID data, the personalization paradigm learn an individual model for each client. Let ω_k denote local model parameters of client k . The local mask $m_k \in \{0, 1\}^{|\omega_k|}$ induces sparsification by $\omega_k \odot m_k$, where $\odot$ is a Hadamard product. Our target is to optimize local sparse models to minimize the average loss of clients, expressed as

$$\begin{aligned} \min_{\omega_1, \dots, \omega_K \in \Omega} \frac{1}{K} \sum_{k=1}^K F_k(\omega_k \odot m_k^*), \\ \text{s.t. } F_k(\omega_k \odot m_k^*) = \mathbb{E}[\mathcal{L}_k(\omega_k \odot m_k^* | \omega; D_k)] \end{aligned} \quad (1)$$

Here, Ω is the feasible parameter space of local models, and $\mathcal{L}_k(\cdot | \omega; D_k)$ denotes the regularization loss of client k over D_k under global parameters ω . For local mask m_k , if an element of m_k is zero, the corresponding parameters are zeroed out, otherwise remain active. In this setting, the nonzero parameters of $\omega_k \odot m_k$ can characterize the submodel. The nonzero parameters are trained and uploaded to the server, meaning that the local running burden and uplink communication volume can be reduced compared to the original dense model. The optimal mask of client k is denoted as m_k^* , which is hard to learn in a non-heuristic way.

Sparse Ratio and Pattern Definition. Let N_k denote the number of nonzero elements in m_k such that $N_k = \|m_k\|_0$. We define the sparse ratio s_k of client k as $s_k = N_k / |\omega_k|$, where a lower ratio implies a higher sparse degree. Given s_k , the local mask m_k is determined by the sparse pattern P_k that indicates the positions of the retained model units. In view of general training speed capabilities, structured sparsification is our primary focus, which considers the structurally indivisible element (e.g., neuron and convolution channel) in DNN as the sparse granularity. We define a network topology element at

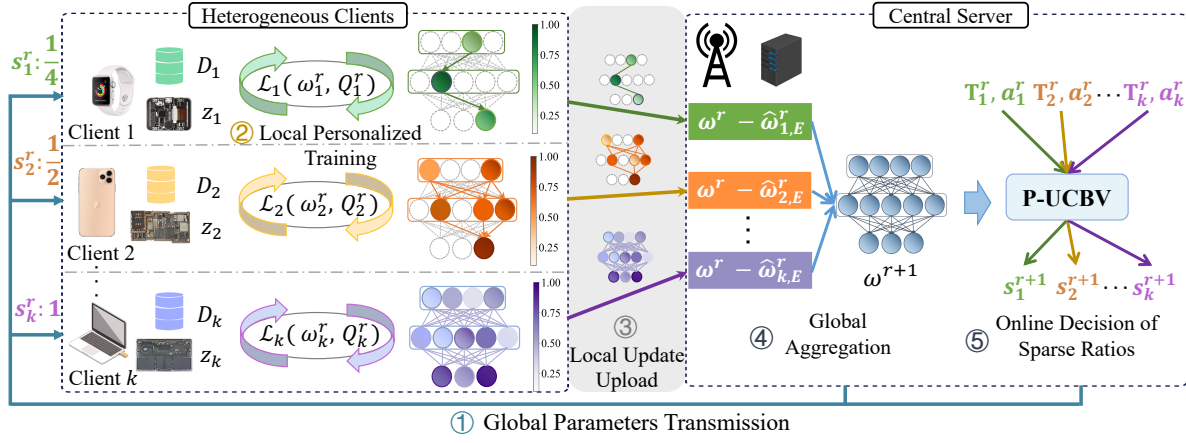


Fig. 2: The overview diagram of our FedLPS framework, where the numbers ①-⑤ represent the training procedures.

the sparse granularity level as a model unit. With unit-wise sparse pattern P_k , the local mask is derived by

$$m_k = \mathcal{M}(P_k | \omega_k, s_k), \quad (2)$$

where $\mathcal{M}(P|\omega, s)$ denotes the construction of a binary mask with the same dimension as ω and sparse ratio s by setting 1 in the positions where P is true and 0 elsewhere. Several studies [28, 29, 39] demonstrate that sparse patterns and ratios have significant impacts on running costs and model performance. In view of non-IID data and diverse resource configurations across edge devices, it is crucial to learn personalized sparse patterns and adaptive sparse ratios, aiming at local resource self-adaption and accuracy improvement in edge computing.

III. THE FEDLPS FRAMEWORK

A. Overview

To deal with complicated issues introduced by non-IID data and system heterogeneity in edge data management and computing scenarios, we propose a learnable sparse customization framework, FedLPS, as illustrated in Figure 2. The core components of FedLPS include (1) *customizing a learnable sparse pattern* by optimizing the importance scores of model units during the back-propagation of local training and (2) *determining an adaptive sparse ratio* through learning the additive feedback of client-specific resource constraints and accuracy changes over local data. Specifically, the procedures of FedLPS in a communication round involve:

- **Local personalized training with learnable sparse patterns:** Each client maintains a unit-wise importance indicator to derive a personalized sparse pattern. The client initializes the local model and sparsifies it with the sparse ratio and pattern. With designed importance-associated loss, local model parameters and importance-based sparse patterns are updated over local data via back propagation. After local training, the client uploads sparse update, local cost, and accuracy to the server.
- **Global aggregation and sparse ratio decision:** The server aggregates local updates and determines sparse ratios. The online decision of sparse ratios is viewed

as a MAB problem. Based on local cost and accuracy statistics, P-UCBV is developed to adaptively select an appropriate sparse ratio for the client.

The above steps are cyclically iterated until round $r \geq R$. The whole process of FedLPS is summarized in Algorithm 1.

B. Learnable Sparse Training to Personalization

Importance Indicator. In non-IID settings, the same model unit exhibits distinct importance on different clients. We introduce an importance indicator for each client to individually measure the significance of each model unit over local data. We define $Q_{k,l}^r$ as the local importance indicator in the l -th iteration of round r on client k , which can be expanded to

$$Q_{k,l}^r = [q_{k,l}^{r,1}, \dots, q_{k,l}^{r,J}]^\top \in \mathbb{R}^J. \quad (3)$$

Here, J denotes the number of sparsifiable units in the local DNN. For instance, in a Fully-Connected Neural Network (FCNN), the sparsifiable units are neurons and J is the number of neurons. Each element $q_{k,l}^{r,j}$, $j \in \{1, \dots, J\}$ denotes the importance score of the j -th sparsifiable unit in the local model. A higher score indicates that the corresponding unit exhibits more significant representation ability for local data. The units with greater significance should be held in sparse models to effectively represent local data and improve performance.

Importance-Derived Sparse Pattern. We can derive a binary sparse pattern $P_{k,l}^r = [\beta_{k,l}^{r,1}, \dots, \beta_{k,l}^{r,J}]^\top \in \{0, 1\}^J$ based on local importance indicator $Q_{k,l}^r$. Given the sparse ratio s_k^r , client k calculates an importance threshold $\tau_{k,l}^r$ as $(1 - s_k^r)$ -quantile of $Q_{k,l}^r$ such that the sparse pattern is formulated by

$$P_{k,l}^r = \gamma(Q_{k,l}^r - \tau_{k,l}^r I), \quad (4)$$

where $\gamma(\cdot)$ is a step function and I denotes a unit vector with the same shape as $Q_{k,l}^r$. If $q_{k,l}^{r,j} < \tau_{k,l}^r$, the j -th element in $P_{k,l}^r$ is set to $\beta_{k,l}^{r,j} = 0$ and the connections corresponding to the j -th model units are masked. In this way, the local mask can be derived in each local iteration l by

$$m_{k,l}^r = \mathcal{M}(P_{k,l}^r | \omega_{k,l}^r, s_k^r) \quad (5)$$

Algorithm 1: FedLPS

1 **Initialize:** client selection fraction ϵ ; the number of local iterations E ; initial global parameters ω^0 ; initial importance indicators $\{Q_1^s, \dots, Q_K^s\}$; initial sparse ratios $\{s_1^0, \dots, s_K^0\}$;
ServerAction:
 1: Set the number of selected clients in a round $C \leftarrow \max(\lfloor \epsilon \cdot K \rfloor, 1)$
 2: **for** each round $r \in \{0, \dots, R-1\}$ **do**
 3: $C_r \leftarrow$ random set of C clients
 4: Send global parameters ω^r and the sparse ratio s_k^r to each selected client $k \in C_r$
 5: **for** each client $k \in C_r$ **in parallel do**
 6: $\hat{\omega}_{k,E}^r, T_k^r, a_k^r \leftarrow$ **ClientUpdate**(ω^r, s_k^r)
 // Local personalized training
 7: **end for**
 8: Global aggregation via Equation (13)
 9: **for** $k \in \{1, 2, \dots, K\}$ **do**
 10: **if** $k \in C_r$ **then**
 11: $s_k^{r+1} \leftarrow$ **P-UCBV**(T_k^r, a_k^r) *// Online decision of sparse ratios by Algorithm 2*
 12: **else**
 13: $s_k^{r+1} \leftarrow s_k^r$
 14: **end if**
 15: **end for**
 16: **end for**

ClientUpdate (ω^r, s_k^r): *// Done by client k*
 17: Initialize $\omega_{k,0}^r \leftarrow \omega^r$ and $Q_{k,0}^r \leftarrow Q_k^s$
 18: **for** each iteration $l = \{0, 1, \dots, E-1\}$ **do**
 19: Sample a batch of training data d_l^k
 20: $\omega_{k,l+1}^r \leftarrow \omega_{k,l}^r - \text{SGD}(\mathcal{L}_k(\omega_{k,l}^r, Q_{k,l}^r | \omega^r, d_l^k), \omega_{k,l}^r \odot \mathcal{M}(\gamma(Q_{k,l}^r - \tau_{k,l}^r I) | \omega_{k,l}^r, s_k^r), \eta_r)$
 21: $Q_{k,l+1}^r = Q_{k,l}^r - \text{SGD}(\mathcal{L}_k(\omega_{k,l}^r, Q_{k,l}^r | \omega^r, d_l^k), Q_{k,l}^r, \eta_r)$
 22: **end for**
 23: Record $Q_k^s \leftarrow Q_{k,E}^r$
 24: Obtain local personalized model with sparse parameters $\hat{\omega}_{k,E}^r \odot \mathcal{M}(\gamma(Q_{k,E}^r - \tau_{k,E}^r I) | \omega_{k,E}^r, s_k^r)$
 25: $\hat{\omega}_{k,E}^r \leftarrow (\omega^r - \omega_{k,E}^r) \odot \mathcal{M}(\gamma(Q_{k,E}^r - \tau_{k,E}^r I) | \omega_{k,E}^r, s_k^r)$
 26: Count local cost T_k^r and average training accuracy a_k^r
 27: Return $\hat{\omega}_{k,E}^r, T_k^r, a_k^r$ to the server

where $\omega_{k,l}^r$ denotes local parameters in current iteration. We expect to tailor a personalized model with a learnable sparse pattern to fit local data. Hence, a personalized importance indicator should be learned to customize the sparse pattern.

Importance-Associated Regularization Loss. We design an importance-associated regularization loss, which consists of three terms. The first one is the task-specific optimization function (e.g., cross-entropy function) between the sparse model prediction $\hat{y}$ and the data label y for any $(x, y) \in D^k$:

$$\mathcal{L}_{tr}^k = \ell(\hat{y}, y | \omega_{k,l}^r \odot \mathcal{M}(\gamma(Q_{k,l}^r - \tau_{k,l}^r I) | \omega_{k,l}^r, s_k^r)). \quad (6)$$

By (6), we establish the correlation between the unit significance $Q_{k,l}^r$ and local data D^k , which enables the importance indicator to be learned through local data mining.

The second one is a local parameter regularization term

$$\mathcal{L}_{pr}^k = \|\omega_{k,l}^r - \omega^r\|^2. \quad (7)$$

It limits local updates not to deviate too much and is commonly used in prior works [40, 41]. The third one is an importance regularization term to prevent excessive shifts and over-sharpening of the importance indicator, formulated as:

$$\mathcal{L}_{ir}^k = \|Q_{k,l}^r - \sigma(|\omega_{k,l}^r|_J)\|^2. \quad (8)$$

Here, $|\omega_{k,l}^r|_J$ is a sum vector of parameter absolute values, with J dimensions. The j -th element of $|\omega_{k,l}^r|_J$ is the sum of the absolute value of parameters corresponding to the j -th unit. $\mathcal{L}_{pr}^k$ has constrained the update of local parameters so that their magnitude sum $|\omega_{k,l}^r|_J$ would not change sharply. $\sigma(\cdot)$ is a sigmoid function, which further smooths $|\omega_{k,l}^r|_J$ and limits it to $[0, 1)$. Hence, (8) avoids $Q_{k,l}^r$ from being too biased or over-sharpened during local training. Combining the three terms, the importance-associated regularization loss is expressed as

$$\mathcal{L}_k(\omega_{k,l}^r, Q_{k,l}^r | \omega^r, D^k) = \mathcal{L}_{tr}^k + \mu * \mathcal{L}_{pr}^k + \lambda * \mathcal{L}_{ir}^k. \quad (9)$$

In this design, we integrate the importance indicator into loss to make unit significance learnable, which assists in updating the importance-based sparse pattern with minimal heuristics.

Client-side Update. With global parameters ω^r and sparse ratio s_k^r , client k measures its computation capability z_k and restricts the sparse model under the carrying capacity by $s_k^r \leq z_k$. If the server-determined sparse ratio exceeds local capability, the client directly resets $s_k^r = z_k$. Consistent with common works [31, 37], we perform layer-wise sparsification and adopt the same sparse ratio s_k^r for each layer. The sparsification is induced into the local initial model via $\omega_{k,0}^r \odot \mathcal{M}(\gamma(Q_{k,0}^r - \tau_{k,0}^r I) | \omega_{k,0}^r, s_k^r)$, where global parameters ω^r are imported into the local model as $\omega_{k,0}^r$. The iterative updates of local parameters and importance-based sparse patterns (lines 18-22 in Algorithm 1) are described as follows. For each local iteration $l = \{0, \dots, E-1\}$, client k constructs the local mask $m_{k,l}^r$ with the importance indicator by (5) and induces sparseness into the local model by $\omega_{k,l}^r \odot m_{k,l}^r$. In back-propagation, local sparse parameters are updated via

$$\omega_{k,l+1}^r = \omega_{k,l}^r - \text{SGD}(\mathcal{L}_k(\omega_{k,l}^r, Q_{k,l}^r | \omega^r, d_l^k), \omega_{k,l}^r, \eta_r), \quad (10)$$

$$\omega_{k,l}^r \odot \mathcal{M}(\gamma(Q_{k,l}^r - \tau_{k,l}^r I) | \omega_{k,l}^r, s_k^r), \eta_r),$$

and the update of the unit-wise importance indicator follows

$$Q_{k,l+1}^r = Q_{k,l}^r - \text{SGD}(\mathcal{L}_k(\omega_{k,l}^r, Q_{k,l}^r | \omega^r, d_l^k), Q_{k,l}^r, \eta_r). \quad (11)$$

Here, $\text{SGD}(\mathcal{L}, Q, \eta)$ denotes the gradient calculation of function $\mathcal{L}$ on Q with a learning rate η , and d_l^k is the training data subset used in the l -th local iteration. The joint optimization of sparse parameters and importance indicators (that derive sparse patterns) enables the learnable sparse training on local specific data, further customizing a sparse model for each client.

Parameter Upload and Global Aggregation. After E iterations, client k locally stores its personalized model $\hat{\omega}_{k,E}^r = \omega_{k,E}^r \odot m_{k,E}^r$ with $m_{k,E}^r = \mathcal{M}(\gamma(Q_{k,E}^r - \tau_{k,E}^r I) | \omega_{k,E}^r, s_k^r)$ (as lines 23-24 in Algorithm 1). The nonzero parameters of residual gradients

$$\hat{\omega}_{k,E}^r = (\omega^r - \omega_{k,E}^r) \odot m_{k,E}^r \quad (12)$$

are uploaded to the server. Subsequently, the server performs global aggregation (line 8 in Algorithm 1) via

$$\omega^{r+1} = \frac{\sum_{k \in \mathcal{C}_r} |D_k| (\omega^r - \hat{\omega}_{k,E}^r)}{\sum_{k \in \mathcal{C}_r} |D_k|}. \quad (13)$$

Note that $\hat{\omega}_{k,E}^r$ denotes the sparse local update, while ω^r represents dense global parameters, which implies that $(\omega^r - \hat{\omega}_{k,E}^r)$ is relatively dense. Furthermore, the local sparse pattern of $\hat{\omega}_{k,E}^r$ is unique for each client k . Thus, the aggregation can provide a relatively dense update for global parameters.

C. Online Decision of Sparse Ratio

In complicated non-IID and system-heterogeneous EDM, the decision of sparse ratios has to consider two aspects. *First, the sparse ratio intuitively determines the model scale and training costs.* The lower sparse ratio contributes to a smaller submodel and fewer costs, which can match lower-tier clients. *Secondly, the sparse ratio is closely correlated with model accuracy over local data* in non-IID settings, where the lower sparse ratio is more likely to deteriorate accuracy. However, most existing works manually set rigid rules, failing to jointly optimize the two aspects. They either directly specify a fixed adjustment rate [33, 42] or simply control sparse ratios according to device capabilities [31, 37], leading to training delays and performance bottlenecks. Thus, it is essential to adaptively determine sparse ratios by learning the additive effect of heterogeneous capabilities and non-IID data for the trade-off between capability adaption and accuracy guarantee.

In the whole FL process, the sparse ratio decision for each client can be regarded as a sequential decision problem, which perfectly matches the modeling of the Multi-Armed Bandit (MAB) problem. The server is viewed as a bandit and arms are feasible sparse ratios. Within limited rounds, the server decides sparse ratios for selected clients (*i.e.*, the bandit chooses arms) in each round. Different sparse ratios bring distinct local costs and training accuracy for each client, which reflect the rewards of arms in our case. We aim at obtaining a sparse ratio sequence for each client to achieve accuracy guarantee with as little training overhead as possible. Thus, we model the ratio decision for each client as a MAB problem, and the server creates K agents to address the MAB problems of K clients. For each agent, the arm space $[0, 1]$ of sparse ratios is infinite.

Although several UCB-extended methods [28, 43] are explored for MAB problems, they either only work with discrete arms without involving the transform of infinite arm space or unilaterally target time saving without considering non-IID data, bringing performance sacrifice and even hindering convergence. In this work, we develop Prompt Upper Confidence Bound Variance (P-UCBV), in which a novel reward function is designed to learn the additive effect of resource restrictions and non-IID data on sparse ratios for the trade-off between resource self-adaption and accuracy improvement. Moreover, we introduce the accuracy-dominated arm elimination, where the sparse ratios that sharply deteriorate accuracy on local data are promptly removed from the feasible arm space to avoid severe accuracy fluctuations and improve decision efficiency.

Algorithm 2: P-UCBV in round r for client $k \in \mathcal{C}_r$

- 1 **Initialize:** partition set $\mathbf{S}_{k,0}$; client selection fraction ϵ ; differential accuracy threshold Δ ; $\varepsilon_0 := 1$; $\xi = R/(K \cdot \epsilon)$; $\psi = \xi/I_0^2$;
 - 2 **Input:** local cost T_k^r ; average training accuracy a_k^r ;
 - 1: $S_k^u \leftarrow$ the partition where s_k^r resides
 - 2: Split S_k^u with s_k^r into $S_k^{u'}$, $S_k^{u''}$ for building $\mathbf{S}_{k,r+1}$
 - 3: **if** $a_k^r - a_k^{r-1} < \Delta$ **then**
 - 4: Remove $S_k^{u'}$ from $\mathbf{S}_{k,r+1}$ // Arm elimination
 - 5: **end if**
 - 6: $\varepsilon_{r+1} \leftarrow \varepsilon_r/2$; $I_{r+1} \leftarrow |\mathbf{S}_{k,r+1}|$;
 - 7: $\psi \leftarrow \xi/I_{r+1}^2$
 - 8: Calculate reward by (15), which is added into the reward lists of partitions $S_k^{u'}$ (if it exists) and $S_k^{u''}$
 - 9: Count the average reward $\{\bar{g}_k^i | i = 1, \dots, I_{r+1}\}$ and variance $\{\bar{v}_k^i | i = 1, \dots, I_{r+1}\}$ of all partitions
 - 10: Select partition $S_k^e = \arg \max_{S_k^i \in \mathbf{S}_{k,r+1}} \mathcal{U}(S_k^i)$
 - 11: Sample s_k^{r+1} from S_k^e
 - 12: **return** s_k^{r+1}
-

Reward Function. Considering the superimposed effect of distinct resource restrictions and non-IID data, both time costs and accuracy changes are involved in the reward function. First, we build a local cost formula involving computation and communication overhead. The local cost of client k in round r is denoted as T_k^r . Let Floating Point Operations (FLOPs) characterize computation overhead and $\hat{F}_k^r$ denote FLOPs of client k in round r . The communication cost is represented by the transmitted parameter size $\hat{B}_k^r$. The local cost T_k^r under client-side resource configurations is calculated by

$$T_k^r = \hat{F}_k^r / F_k^r + \alpha \hat{B}_k^r / B_k^r, \quad (14)$$

where F_k^r and B_k^r are maximum capacities of locally available computation and bandwidth. The cost calculation of (14) is implemented on the client, meaning that privacy-sensitive configuration information (*e.g.*, computing power F_k^r) of clients will not be leaked. The reward of sparse ratio s_k^r is defined as

$$G(s_k^r) = (U(a_k^r) - U(a_k^{r-1})) / T_k^r. \quad (15)$$

The utility function $U(\cdot)$ [44] is used to moderately transform the accuracy, which accounts for marginal accuracy gains near the end of FedLPS.

P-UCBV Algorithm. The detailed process of P-UCBV is described in Algorithm 2. For each agent, the infinite ratio space is divided based on the decision tree [28]. Initially, agent k holds I_0 ratio partitions $\mathbf{S}_{k,0} = \{S_k^1, \dots, S_k^{I_0}\}$ with $\bigcup_{i=1}^{I_0} S_k^i = [0, 1]$ and randomly chooses a partition to sample the initial sparse ratio s_k^0 from the selected partition. We evaluate the initial global model on local data to obtain the original accuracy set $\{a_k^{-1} | k = 1, \dots, K\}$. In each round r , the server splits the last selected partition S_k^u (*i.e.*, $s_k^r \in S_k^u$) into two partitions $S_k^{u'}$ and $S_k^{u''}$, where s_k^r is the split point (line 2). If s_k^r causes a larger sacrifice in accuracy, the accuracy-dominated prompt arm-elimination operation is activated, and the partition $S_k^{u'}$ is promptly removed to build a new partition set $\mathbf{S}_{k,r+1}$ (lines 3-5). Subsequently, the reward of the selected

sparse ratio s_k^r is calculated by Equation (15) (line 8). We count the average reward of the i -th arm (*i.e.*, S_k^i) as

$$\bar{g}_k^i = \frac{1}{h_k^i} \sum_{\ell=1}^{h_k^i} G_{k,\ell}^i, \quad (16)$$

where $G_{k,\ell}^i$ is the reward feedback when S_k^i is chosen for the ℓ -th time, and h_k^i is pulled times of S_k^i . The variance is calculated by $\bar{v}_k^i = \frac{1}{h_k^i} \sum_{\ell=1}^{h_k^i} (G_{k,\ell}^i - \bar{g}_k^i)^2$. With $\bar{g}_k^i$ and $\bar{v}_k^i$, we compute the UCBV value of each partition $S_k^i \in \mathbf{S}_{k,r+1}$ by

$$U(S_k^i) = \bar{g}_k^i + \sqrt{\frac{\rho(\bar{v}_k^i + 2) \log(\xi \psi \varepsilon_{r+1})}{4(h_k^i + 1)}}, \quad (17)$$

where $\xi = R/(K \cdot \epsilon)$ and ρ is a preset constant. ε_{r+1} and ψ are updated in P-UCBV as lines 6-7. Then, the optimal partition is selected to sample a new sparse ratio (lines 10-11).

IV. FURTHER ANALYSIS

A. Cost Analysis

Local Computation Cost. We analyze the local computation costs of FedLPS, where the updates of sparse parameters and importance indicators require computing support. The nonzero parameters of the sparse model are locally trained, significantly alleviating the local computation burden compared to the original dense model. On the other hand, the computation cost for the importance indicator updating is much less than that of model parameters, which can be ignored. Because the size of a unit-wise importance indicator is much smaller than the size of model parameters. Considering a model consisting of three fully-connected layers with 1024 neurons, the FLOPs of updating the local model are 15.36×10^5 in an iteration, following the FLOP calculation in [45]. While the FLOPs of updating importance indicators are 750, which is much less than 15.36×10^5 and even can be ignored. Besides, sparse ratios are decided on the server with adequate computing power, without involving local computation consumption. Hence, FedLPS effectively reduces local computation costs.

Communication Cost. For uplink communication, only nonzero parameters of local sparse models and tiny unit-wise binary patterns are uploaded, which can significantly mitigate uplink overhead. In terms of downlink communication, the server delivers relatively dense global parameters and selected sparse ratios (much less than global parameters and even can be ignored) to clients. In this way, FedLPS has a similar downlink overhead as conventional FL frameworks (*e.g.*, FedAvg).

Global Cost. FedLPS adopts synchronous aggregation such that the global time cost is determined by the slowest client. In round r , the global time cost is modeled by

$$T^r = \max_{k \in \mathcal{C}_r} T_k^r, \quad (18)$$

where T_k^r is calculated by (14). Based on the above analysis, FedLPS can mitigate client-side computation burden (*i.e.*, $\hat{F}_k^r$) and uplink communication volume (*i.e.*, $\hat{B}_k^r$) such that local time costs $T_k^r, k \in \mathcal{C}_r$ are also reduced. When T_k^r

decreases, the corresponding global cost T^r will be reduced. Our experiments have demonstrated that FedLPS reduces the total time cost to accelerate training, as shown in Section V.

B. Privacy Analysis

FedLPS transmits residual model parameters without sharing privacy-sensitive raw data, similar to conventional FL frameworks, which mitigates data privacy concerns in edge networks. For transmitting local cost and training accuracy, previous work [28] has verified its reliability and practicality. As mentioned in [28], the server needs to be online aware of the different capabilities of clients in heterogeneous FL settings, while the decision-making strategy based on transmitted local time costs and training evaluation results can avoid the direct leakage of privacy-sensitive computing power information, practical in heterogeneous edge computing.

Furthermore, FedLPS is orthogonal to existing FL privacy-preserving techniques (*e.g.*, differential privacy [46] and homomorphic encryption [47]), which can be directly applied with FedLPS. For instance, we could add noise to transmitted parameters via differential privacy, and encrypt local costs and training accuracy to further guarantee privacy security.

C. Convergence Analysis

In this section, we provide a formal theoretical analysis to guarantee the convergence of FedLPS under Stochastic Gradient Descent (SGD) optimization with the constraints of learning rates. Let $\tilde{\omega}_{k,l}^r = \omega_{k,l}^r \odot m_{k,l}^r$ with $m_{k,l}^r = \mathcal{M}(\gamma(Q_{k,l}^r - \tau_{k,l}^r I) | \omega_{k,l}^r, s_k^r)$ and $\nabla_{\tilde{\omega}} \mathcal{L}_k(\tilde{\omega}_{k,l}^r, Q_{k,l}^r; \xi_l) = 1/\eta_r \cdot \text{SGD}(\mathcal{L}_k(\omega_{k,l}^r, Q_{k,l}^r | \omega^r, d_l^k), \omega_{k,l}^r \odot \mathcal{M}(\gamma(Q_{k,l}^r - \tau_{k,l}^r I) | \omega_{k,l}^r, s_k^r), \eta_r)$ denotes the estimate of $\nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l}^r)$, where ξ_l is a random variable. Besides, we define the optimal local mask of client k as m_k^* .

To facilitate analysis, we make the following assumptions.

Assumption 1 (L-Lipschitz Smoothness). For each client $k \in \{1, \dots, K\}$, the function F_k is smooth such that

$$\|\nabla_{\omega} F_k(\omega) - \nabla_{\omega} F_k(\omega')\| \leq L \|\omega - \omega'\|.$$

Assumption 2 (Bounded Gradient Estimator Bias). The stochastic gradient estimation of each client $k \in \{1, \dots, K\}$ satisfies σ^2 -bounded bias with

$$\mathbb{E} \|m_{k,l}^r \odot \nabla_{\tilde{\omega}} \mathcal{L}_k(\tilde{\omega}_{k,l}^r, Q_{k,l}^r; \xi_l) - m_{k,l}^r \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l}^r)\|^2 \leq \sigma^2$$

for all $r \in \{0, \dots, R-1\}$ and $l \in \{0, \dots, E-1\}$.

Assumption 3 (Bounded Local Sparse Gradient). For each client $k \in \{1, \dots, K\}$, the expected squared norm of local sparse gradients is bounded by H^2 , *i.e.*,

$$\mathbb{E} \|m_{k,l}^r \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l}^r)\|^2 \leq H^2.$$

Assumption 4 (Bounded Sparse Gradient Distance). For each client $k \in \{1, \dots, K\}$, the distance between local sparse gradients with the optimal mask and average sparse gradients of all clients are bounded by a constant B with

$$\|m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_k) - \frac{1}{K} \sum_{i=1}^K m_i^* \odot \nabla_{\tilde{\omega}} F_i(\tilde{\omega}_i)\| \leq B,$$

where $\tilde{\omega}_k = m_k^* \odot \omega_k$ for any $\omega_k \in \Omega$.

Assumptions 1-3 are common in many FL convergence studies [48–50]. Assumption 4 bounds the sparse gradient difference between the local loss and global average loss, which is widely used to characterize client diversities [51, 52]. Based on these assumptions, we analyze the convergence of FedLPS. In round r , a uniform learning rate η_r is adapted for all selected clients and the local parameter update in the l -th iteration can be rewritten as $\omega_{k,l+1}^r = \omega_{k,l}^r - \eta_r \cdot m_{k,l}^r \odot \nabla_{\tilde{\omega}} \mathcal{L}_k(\tilde{\omega}_{k,l}^r, Q_{k,l}^r; \xi_l)$. Firstly, we give the upper bound of the gap between global and local parameters, as shown in Lemma 1.

Lemma 1. *Let Assumptions 1-4 hold. For any $r \in \{0, \dots, R-1\}$, $l \in \{0, \dots, E-1\}$, it follows*

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \|\omega_{k,l}^r - \omega^{r+1}\|^2 \leq 5E\eta_r^2(\sigma^2 + 6EB^2 + 18EH^2)$$

with the learning rate $\eta_r \leq \sqrt{\frac{1}{24ERV_rL^2}}$, where $V_r = \max_{k \in [K], l \in [E]} \left\{ \frac{\|m_{k,E}^r \odot m_{k,l}^r \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l}^r) - m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r})\|^2}{\|\nabla_{\tilde{\omega}} F_k(\omega_{k,l}^r) - \nabla_{\tilde{\omega}} F_k(\omega^{r+1})\|^2} \right\}$ with $\tilde{\omega}_{k,r} = m_k^* \odot \omega_{k,0}^r$ and $\tilde{\omega}_{k,l}^r = m_{k,l}^r \odot \omega_{k,l}^r$.

Proof. To prove the above result, we define $q_{k,l}^r(\tilde{\omega}_{k,l}^r) = \nabla_{\tilde{\omega}} \mathcal{L}_k(\tilde{\omega}_{k,l}^r, Q_{k,l}^r; \xi_l)$ and specify that

- $A_1 = m_{k,l-1}^r \odot q_{k,l-1}^r(\tilde{\omega}_{k,l-1}^r) - m_{k,l-1}^r \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l-1}^r)$
- $A_2 = m_{k,l-1}^r \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l-1}^r) - m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r})$
- $A_3 = m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r}) - \frac{1}{K} \sum_{i=1}^K m_i^* \odot \nabla_{\tilde{\omega}} F_i(\tilde{\omega}_{i,r})$
- $A_4 = \frac{1}{K} \sum_{k=1}^K m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r})$

In this way, there is

$$\begin{aligned} \mathbb{E} \|\omega_{k,l}^r - \omega^{r+1}\|^2 &= \mathbb{E} \|\omega_{k,l-1}^r - \omega^{r+1} - \eta_r m_{k,l-1}^r \odot q_{k,l-1}^r(\tilde{\omega}_{k,l-1}^r)\|^2 \\ &\leq \left(1 + \frac{1}{2E-1}\right) \mathbb{E} \|\omega_{k,l-1}^r - \omega^{r+1}\|^2 + \eta_r^2 \mathbb{E} \|A_1\|^2 \\ &\quad + 6E\eta_r^2 \mathbb{E} \|A_2\|^2 + 6E\eta_r^2 \mathbb{E} \|A_3\|^2 + 6E\eta_r^2 \mathbb{E} \|A_4\|^2. \end{aligned} \quad (19)$$

Bounding A_1 by Assumption 2, it holds

$$\eta_r^2 \mathbb{E} \|m_{k,l-1}^r \odot (q_{k,l-1}^r(\tilde{\omega}_{k,l-1}^r) - \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l-1}^r))\|^2 \leq \eta_r^2 \sigma^2.$$

In terms of A_2 , we observe that

$$\begin{aligned} 6E\eta_r^2 \mathbb{E} \|m_{k,l-1}^r \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l-1}^r) - m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r})\|^2 \\ \leq 12EV_r\eta_r^2 \mathbb{E} \|\nabla_{\tilde{\omega}} F_k(\omega_{k,l-1}^r) - \nabla_{\tilde{\omega}} F_k(\omega^{r+1})\|^2 + 12EH^2\eta_r^2 \\ \leq 12EV_rL^2\eta_r^2 \mathbb{E} \|\omega_{k,l-1}^r - \omega^{r+1}\|^2 + 12EH^2\eta_r^2, \end{aligned} \quad (20)$$

where the last inequality holds by Assumption 1.

Bounding A_3 with Assumption 4, there exists

$$\begin{aligned} 6E\eta_r^2 \mathbb{E} \left\| m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r}) - \frac{1}{K} \sum_{i=1}^K m_i^* \odot \nabla_{\tilde{\omega}} F_i(\tilde{\omega}_{i,r}) \right\|^2 \\ \leq 6E\eta_r^2 B^2. \end{aligned} \quad (21)$$

For A_4 , due to Assumption 3 and Lemma 1 in [34], it satisfies

$$6E\eta_r^2 \mathbb{E} \left\| \frac{1}{K} \sum_{k=1}^K m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r}) \right\|^2 \leq 6E\eta_r^2 H^2. \quad (22)$$

Combining the four terms together, we obtain the following:

$$\begin{aligned} \mathbb{E} \|\omega_{k,l}^r - \omega^{r+1}\|^2 &\leq \left(1 + \frac{1}{2E-1}\right) \mathbb{E} \|\omega_{k,l-1}^r - \omega^{r+1}\|^2 + \eta_r^2 \sigma^2 \\ &\quad + 6E\eta_r^2(B^2 + 3H^2) + 12EV_rL^2\eta_r^2 \mathbb{E} \|\omega_{k,l-1}^r - \omega^{r+1}\|^2. \end{aligned} \quad (23)$$

Considering the average of all clients, it follows

$$\begin{aligned} \frac{1}{K} \sum_{k=1}^K \mathbb{E} \|\omega_{k,l}^r - \omega^{r+1}\|^2 &\leq \left(1 + \frac{1}{E-1}\right) \frac{1}{K} \sum_{k=1}^K \mathbb{E} \|\omega_{k,l-1}^r - \omega^{r+1}\|^2 \\ &\quad + \eta_r^2 \sigma^2 + 6E\eta_r^2(B^2 + 3H^2), \end{aligned} \quad (24)$$

where the first inequality holds because $\eta_r \leq \sqrt{\frac{1}{24ERV_rL^2}}$ such that $\frac{1}{2E-1} + 12EV_rL^2\eta_r^2 \leq \frac{1}{2E-1} + \frac{1}{2R} \leq \frac{1}{E-1}$ with $R \geq E-1$ and $E > 1$. **Expanding the recursion**, we get

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \|\omega_{k,l}^r - \omega^{r+1}\|^2 \leq 5E\eta_r^2(\sigma^2 + 6EB^2 + 18EH^2),$$

concluding the proof of Lemma 1. $\square$

Based on Lemma 1, we can provide the convergence result of FedLPS in Theorem 1.

Theorem 1. *Let Assumptions 1-4 hold. Choose $\phi = 4\sqrt{6}L \max_{r \in [R]} \sqrt{V_r}$, $\varphi = \max_{r \in [R]} \sqrt{\frac{E}{6V_r}}$, and the learning rate $\eta_r \leq \sqrt{\frac{1}{24ERV_rL^2}}$. Then for FedLPS, it follows*

$$\begin{aligned} \frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E} \left\| \frac{1}{K} \sum_{k=1}^K \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r}) \right\|^2 &\leq \frac{\phi}{\sqrt{ER}}(f_0 - f^*) + \frac{\varphi}{\sqrt{R}} \left(2H^2 + \frac{\sigma^2}{KE}\right) \\ &\quad + \frac{1}{R} \left(\frac{5}{24} + \frac{5\varphi}{12\sqrt{R}}\right) (\sigma^2 + 6EB^2 + 18EH^2), \end{aligned} \quad (25)$$

where $f_0 = \frac{1}{K} \sum_{k=1}^K F_k(\tilde{\omega}_{k,0})$ and $f^* = \frac{1}{K} \sum_{k=1}^K F_k(\tilde{\omega}_k^*)$ with the optimal local sparse parameters $\tilde{\omega}_k^*$.

Proof. To verify the above theorem, we define $f(\omega^r) = \frac{1}{K} \sum_{k=1}^K F_k(m_k^* \odot \omega^r) = \frac{1}{K} \sum_{k=1}^K F_k(\tilde{\omega}_{k,r})$ since $\omega_{k,0}^r = \omega^r$ and $\nabla f(\omega^r) = \frac{1}{K} \sum_{k=1}^K \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r})$. By Assumption 1 and Lemma 5 in [34], there is

$$f(\omega^{r+1}) \leq f(\omega^r) - \langle \nabla f(\omega^r), \omega^{r+1} - \omega^r \rangle + \frac{L}{2} \|\omega^{r+1} - \omega^r\|^2. \quad (26)$$

For the second term on the right of (26), we derive that

$$\begin{aligned} -\mathbb{E} [\langle \nabla f(\omega^r), \omega^{r+1} - \omega^r \rangle] &\leq \frac{\eta_r}{2E} \left(\mathbb{E} \left\| \frac{1}{K} \sum_{k=1}^K \sum_{l=0}^{E-1} [m_{k,E}^r \odot m_{k,l}^r \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l}^r) \right. \right. \\ &\quad \left. \left. - E \nabla f(\omega^r)] \right\|^2 - E^2 \mathbb{E} \|\nabla f(\omega^r)\|^2 \right) \\ &\leq \frac{\eta_r V_r L^2}{2K} \sum_{k=1}^K \sum_{l=0}^{E-1} \mathbb{E} \|\omega_{k,l}^r - \omega^{r+1}\|^2 - \frac{\eta_r E}{2} \mathbb{E} \|\nabla f(\omega^r)\|^2. \end{aligned} \quad (27)$$

For the third term on the right of (26), it holds:

$$\begin{aligned}
& \frac{L}{2} \mathbb{E} \|\omega^{r+1} - \omega^r\|^2 \\
& \leq L\eta_r^2 \mathbb{E} \left\| \frac{1}{K} \sum_{k=1}^K m_{k,E}^r \odot \sum_{l=0}^{E-1} m_{k,l}^r \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,l}^r) \right\|^2 \\
& \quad + \frac{LE\eta_r^2\sigma^2}{K} \\
& \leq \frac{2LEV_r\eta_r^2}{K} \sum_{l=0}^{E-1} \sum_{k=1}^K \|\nabla_{\omega} F_k(\omega_{k,l}^r) - \nabla_{\omega} F_k(\omega^{r+1})\|^2 \\
& \quad + 2LE^2\eta_r^2 \mathbb{E} \left\| \frac{1}{K} \sum_{k=1}^K m_k^* \odot \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r}) \right\|^2 + \frac{LE\eta_r^2\sigma^2}{K} \\
& \leq \frac{2L^3EV_r\eta_r^2}{K} \sum_{l=0}^{E-1} \sum_{k=1}^K \mathbb{E} \|\omega_{k,l}^r - \omega^{r+1}\|^2 + 2LE^2H^2\eta_r^2 \\
& \quad + \frac{LE\eta_r^2\sigma^2}{K}. \tag{28}
\end{aligned}$$

Combining (27) and (28), we can obtain the following:

$$\begin{aligned}
\mathbb{E}[f(\omega^{r+1})] & \leq \mathbb{E}[f(\omega^r)] - \frac{\eta_r E}{2} \mathbb{E} \|\nabla f(\omega^r)\|^2 \\
& \quad + \frac{5E^2L^2V_r\eta_r^3}{2} (\sigma^2 + 6EB^2 + 18EH^2) \\
& \quad + 10E^3L^3V_r\eta_r^4 (\sigma^2 + 6EB^2 + 18EH^2) \\
& \quad + 2E^2LH^2\eta_r^2 + \frac{LE\eta_r^2\sigma^2}{K}. \tag{29}
\end{aligned}$$

Furthermore, we transform inequality (29) into

$$\begin{aligned}
& \mathbb{E} \|\nabla f(\omega^r)\|^2 \\
& \leq \frac{2\{\mathbb{E}[f(\omega^r)] - \mathbb{E}[f(\omega^{r+1})]\}}{\eta_r E} + 2EL\eta_r \left(2H^2 + \frac{\sigma^2}{KE} \right) \\
& \quad + 5EL^2V_r\eta_r^2 (1 + 4EL\eta_r) (\sigma^2 + 6EB^2 + 18EH^2). \tag{30}
\end{aligned}$$

By $\eta_r \leq \sqrt{\frac{1}{24ERV_rL^2}}$, there is

$$\begin{aligned}
\mathbb{E} \|\nabla f(\omega^r)\|^2 & \leq \frac{4\sqrt{6RV_r}L\{\mathbb{E}[f(\omega^r)] - \mathbb{E}[f(\omega^{r+1})]\}}{\sqrt{E}} \\
& \quad + \frac{5}{24R} \left(1 + \frac{2\sqrt{E}}{\sqrt{6RV_r}} \right) (\sigma^2 + 6EB^2 + 18EH^2) \\
& \quad + \frac{\sqrt{E}}{\sqrt{6RV_r}} \left(2H^2 + \frac{\sigma^2}{KE} \right). \tag{31}
\end{aligned}$$

Let $\phi = 4\sqrt{6}L \max_{r \in [R]} \sqrt{V_r}$ and $\varphi = \max_{r \in [R]} \sqrt{\frac{E}{6V_r}}$. By $\nabla f(\omega^r) = \frac{1}{K} \sum_{k=1}^K \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r})$, it holds:

$$\begin{aligned}
& \frac{1}{R} \sum_{r=0}^{R-1} \mathbb{E} \left\| \frac{1}{K} \sum_{k=1}^K \nabla_{\tilde{\omega}} F_k(\tilde{\omega}_{k,r}) \right\|^2 \\
& \leq \frac{\phi}{\sqrt{ER}} (f_0 - f^*) + \frac{\varphi}{\sqrt{R}} \left(2H^2 + \frac{\sigma^2}{KE} \right) \\
& \quad + \frac{5}{24R} \left(1 + \frac{2\varphi}{\sqrt{R}} \right) (\sigma^2 + 6EB^2 + 18EH^2). \tag{32}
\end{aligned}$$

Thus, Theorem 1 can be proven. $\square$

V. EXPERIMENT EVALUATION

We conduct extensive experiments on the classic datasets and models for image classification and next-word prediction tasks, aiming at answering the following questions:

- **Q1:** Can FedLPS offer higher accuracy with fewer costs compared to baselines in dual-heterogeneous settings?
- **Q2:** Does FedLPS perform better in convergence?
- **Q3:** How do different non-IID and system-heterogeneous levels affect the performance of FedLPS?
- **Q4:** How do the learnable patterns and adaptive ratios obtained by P-UCBV affect performance, respectively?

A. Experimental Settings

Datasets. We adopt 5 benchmarks to evaluate the performance of FedLPS: MNIST [53], CIFAR10, CIFAR100 [54], Tiny-Imagenet [55], and Reddit [56]. Reddit is a realistic federated dataset for the next-word prediction task, which contains a large number of English comment texts from real users [56]. We adopt the top 100 users with more data as clients, where different clients have different sample sizes. Considering different speaking preferences, Reddit is inherently non-IID. The other datasets are widely used for image classification. MNIST and CIFAR10 involve 10-class images. For CIFAR100, there are 100-class images. Tiny-Imagenet contains 100k color images from 200 classes. We utilize the pathological partition strategy [45] to obtain highly non-IID data, where each client is randomly assigned 2 classes for MNIST and CIFAR10, 10 classes for CIFAR100, and 20 classes for Tiny-Imagenet.

Models. A CNN model with two convolutional layers is used for MNIST, and the VGG11 [57] is adopted as the backbone of CIFAR10. CIFAR100 and Tiny-Imagenet are trained on VGG13 and VGG16. We employ a RNN model with two LSTM layers and a softmax layer [56] for Reddit.

Baselines. We compare against various FL frameworks, including (1) *conventional FL*, (2) *heterogeneous sparse-training FL*, and (3) *personalized FL*. For *conventional FL* frameworks with the same size models on all clients, we consider that:

- *FedAvg* [48] and *FedProx* [25] are classic FL frameworks, which require clients to locally train dense models and upload all updates for average aggregation.
- *Oort* [23] and *REFL* [24] explore intelligent client selection in heterogeneous FL. Oort ignores local data diversity, while REFL alleviates the diversity issue by diversity measure and staleness-aware aggregation, where the stale updates negatively impact accuracy and convergence.
- *PruneFL* [50] and *CS* [38] are state-of-the-art FL sparsification methods, where clients inherit the same sparse ratio and train submodels with the same sizes. PruneFL requires a powerful client for initial dense model sparsification and then distributes the sparse model to all clients for joint learning. CS applies unstructured sparsification in FL, limited by the specialized hardware requirement.

$\square$ In terms of *heterogeneous sparse-training FL* frameworks,

TABLE I: Accuracy and FLOPs results of different methods.

Methods	MNIST		CIFAR10		CIFAR100		Tiny-Imagenet		Reddit	
	Acc (%)	FLOPs (1e12)	Acc (%)	FLOPs (1e12)	Acc (%)	FLOPs (1e12)	Acc (%)	FLOPs (1e12)	Acc (%)	FLOPs (1e12)
FedAvg	88.15±1.14	2.8	31.28±1.24	825.4	24.30±0.53	4934.3	5.21±0.14	27081.0	23.39±0.56	10.5
FedProx	87.53±1.16	2.8	31.12±1.38	825.4	24.83±0.52	4934.3	5.66±0.12	27081.0	23.37±0.57	10.5
Oort	95.21±0.42	2.8	33.62±0.97	825.4	26.56±0.39	4934.3	4.93±0.12	26539.4	23.91±0.06	10.5
REFL	94.67±0.33	1.3	33.10±1.68	642.1	24.87±0.64	2950.7	5.03±0.10	15436.2	24.45±0.07	8.8
PruneFL	90.90±0.34	2.2	32.20±1.66	793.3	26.89±0.46	4673.8	5.03±0.13	26076.9	23.60±0.19	9.0
CS	92.44±1.16	1.4	28.09±0.84	417.0	27.76±0.61	2493.7	6.02±0.10	13686.2	21.69±0.24	5.3
eFD	93.31±0.91	1.1	28.44±0.98	380.1	22.74±0.45	1989.5	3.85±0.11	10337.5	23.20±0.04	4.6
Fjord	92.56±1.03	1.1	28.04±1.27	380.1	25.83±0.77	1989.5	3.77±0.08	10337.5	23.09±0.56	4.6
HeteroFL	90.76±1.05	1.1	33.22±1.73	380.1	26.34±0.77	1989.5	4.65±0.11	10337.5	23.38±0.11	4.6
FedRolex	93.73±1.44	1.1	33.59±1.91	380.1	26.28±1.02	1989.5	4.82±0.12	10337.5	23.84±0.25	4.6
FedMP	91.63±0.90	1.7	28.99±0.78	532.0	30.36±0.95	2373.5	5.40±0.16	16019.2	23.57±0.87	6.3
DepthFL	95.44±0.53	2.2	29.72±1.06	725.8	20.41±0.52	3431.6	3.83±0.13	23565.1	23.11±0.16	4.9
Ditto	92.00±0.30	5.6	82.60±0.29	1650.8	49.01±0.19	9868.6	11.92±0.04	54162.0	23.97±0.24	21.0
FedPer	93.45±0.07	2.8	79.76±0.15	825.4	60.43±0.08	4934.3	14.96±0.10	27081.0	24.64±0.06	10.5
FedRep	91.92±0.30	2.8	80.87±0.09	825.4	45.36±0.53	4934.3	15.18±0.06	27081.0	24.43±0.07	10.5
Per-FedAvg	93.46±0.33	2.8	85.00±0.09	660.3	58.57±1.57	3256.6	16.60±0.51	18956.7	24.65±0.42	9.7
LotteryFL	93.37±1.28	1.8	75.71±1.00	656.7	44.83±0.58	3276.9	11.11±0.21	16454.3	24.33±0.02	6.2
Hermes	94.05±0.60	1.7	81.07±1.58	565.9	47.09±0.81	3547.7	11.52±0.10	17168.1	24.89±0.10	6.1
FedSpa	93.94±0.85	1.4	78.83±2.41	522.9	55.76±1.02	3057.5	9.20±0.12	16229.8	24.62±0.04	5.3
FedP3	92.41±0.34	1.0	78.46±0.16	320.1	54.20±0.19	1951.6	17.69±0.05	13093.1	23.91±0.09	4.6
FedLPS	96.77±0.11	0.8	87.43±0.19	268.5	62.80±0.05	1706.7	21.48±0.12	9307.3	26.17±0.04	4.2

- *Fjord* [37], *HeteroFL* [31], and *FedRolex* [27] directly select sparse ratios based on local capabilities and prune model units in an ordered manner.
- *DepthFL* [58] tailors local models by removing the deepest layers under the resource-based sparse ratios.
- *FedMP* [28] explores extended UCB to select sparse ratio and then prunes model units based on magnitude.

Among *personalized FL* methods, we compare that:

- *FedPer* [59] and *FedRep* [21] view the last layers of local models as personalized modules that are not uploaded.
- *Ditto* [40] introduces regularization into the local training for robust personalization.
- *PerFedAvg* [60] studies the personalized variant of FedAvg within model-agnostic meta-learning.
- *LotteryFL* [42] and *Hermes* [33] gradually decline sparse ratios from 1 at a fixed rate and prune a fixed number of low-magnitude weights to customize local sparse models.
- *FedSpa* [34] introduces dynamic sparse training into FL to learn an always-sparse personalized model with a uniform sparse ratio for each client.
- *FedP3* [35] integrates global and local pruning strategies and allows personalization based on the client resource constraints, without involving pattern optimization.

Configurations. For heterogeneous clients, we consider five capability levels $z_k \in \{1, 1/2, 1/4, 1/8, 1/16\}$ and uniformly sample from possible levels for K clients, referring to [31]. We set the optimal device (*i.e.*, $z_k = 1$) with Adreno 630, which has the computation capability of 727G floating-point operations per second [61]. During the training, the local available resources can dynamically change, since users also have other irregular task requirements that may bring the changes of available power. The total number of clients is $K = 100$ for

MNIST and Reddit, and $K = 50$ is set for other datasets. The number of communication rounds is $R = 100$. In each round, the server randomly selects 10 clients. During local training, the batch size is set to 20. SGD optimizer is adopted with 0.1 learning rate for image classification tasks and 8 for next-word prediction, where gradient clip is involved as referring to [56]. For (9), we set $\mu = 1$ and $\lambda = 1$. The utility function used in (15) is defined to be $U(x) = 10 - \frac{20}{1+e^{(0.35x)}}$. Our code is available at <https://github.com/sunnyxuejij/FedLPS>.

Evaluation Metrics. We adopt test accuracy and FLOPs as the main evaluation metrics. For FLOPs, many FL works [34, 45, 50] use it to characterize local computation costs, where fewer FLOPs mean less training overhead. Generally, FLOPs and running time are positively correlated in certain settings [50, 62]. The total time cost is also evaluated in our experiments.

B. Performance Comparison (Q1)

Table I summarizes experimental results, where *Acc* means the average accuracy of all clients on local test data, and *FLOPs* denote total floating operations of all clients during the FL process. It is remarkable that FedLPS consistently achieves the start-of-the-art accuracy with minimal computation costs.

First, conventional FL frameworks FedAvg and FedProx generally perform poorly on all datasets. Although Oort and REFL mitigate the heterogeneity bottleneck by adaptive client selection, they still suffer from accuracy degradation in non-IID settings. Compared to REFL, FedLPS achieves 1.72%-54.33% accuracy gain with 38.46%-58.18% FLOP reduction. Besides, recent sparse works (based on conventional FL) PruneFL and CS also reduce local FLOPs, but still show lower accuracy. Compared to CS, FedLPS improves accuracy by 4.33%-59.34% while reducing 20.75%-42.86% FLOP costs.

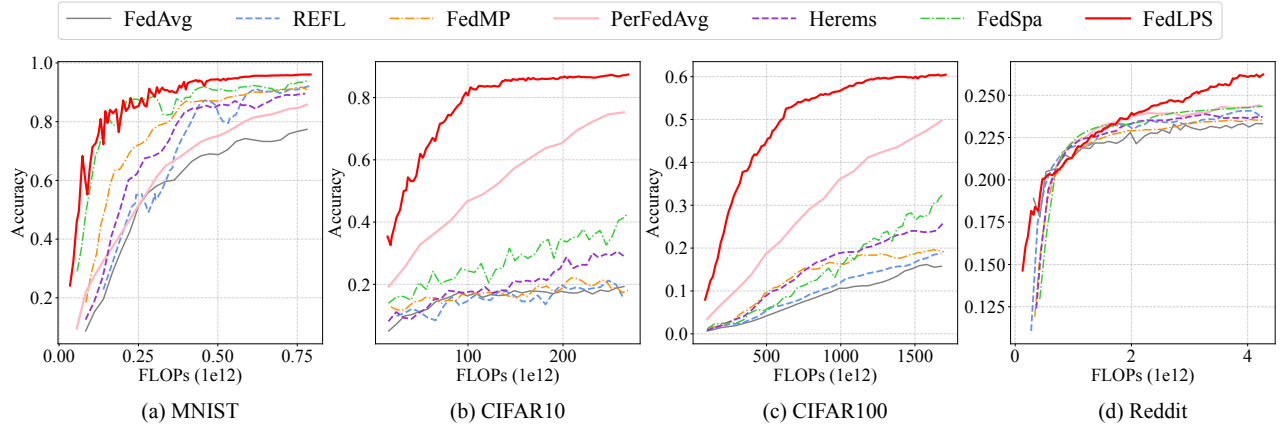


Fig. 3: Test accuracy versus FLOP computation costs on four datasets.

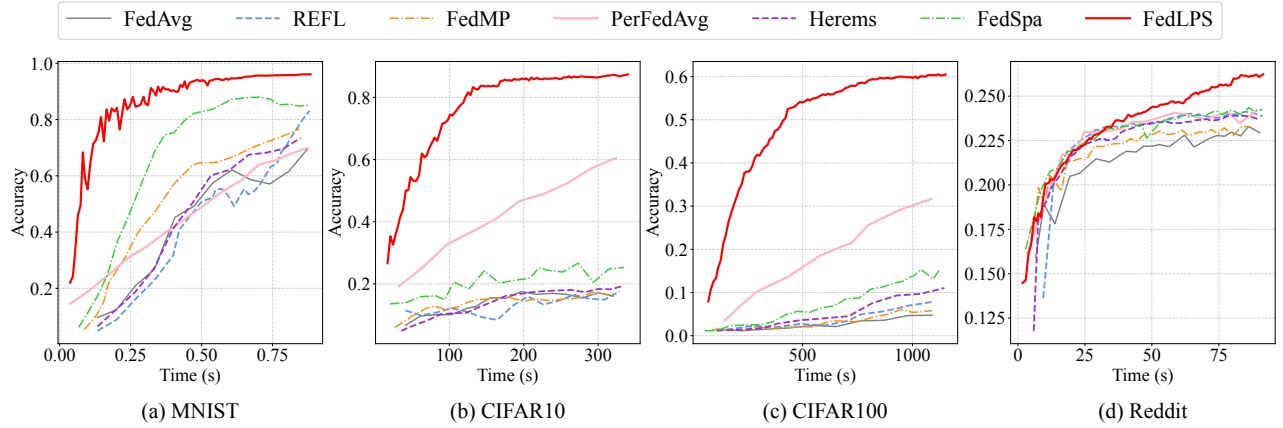


Fig. 4: Test accuracy versus running time on four datasets.

Second, heterogeneous sparse FL methods significantly outperform conventional FL. A major merit is that they assign submodels with different sizes based on client-side computing power. However, they finally deploy a shared inference model on all clients, resulting in lower accuracy in non-IID settings. In particular, FedLPS outperforms Hermes in test accuracy and training costs, which enhances accuracy by 1.28%-15.71% and reduces computation costs by more than 30%. Compared to the advanced FedRolex, FedLPS achieves 2.33%-53.84% accuracy gains and saves up to 29% FLOP computation costs.

Furthermore, personalized FL methods tailor a local model for each client to fit non-IID data and yield better inference accuracy. Combined with sparsification techniques, LotteryFL, Herems, FedSpa, and FedP3 learn personalized sparse models with fewer computation costs. Finally, FedLPS outperforms state-of-the-art personalized FL methods in inference accuracy and computation overhead on all datasets. Specifically, FedLPS provides 1.28%-4.23% accuracy gains with up to 60% reduction of FLOPs.

C. Convergence and Running Time Evaluation (Q1 and Q2)

To evaluate the convergence of FedLPS, we report the test accuracy varying with total FLOPs and running time. As shown in Figure 3, FedLPS generally offers higher ac-

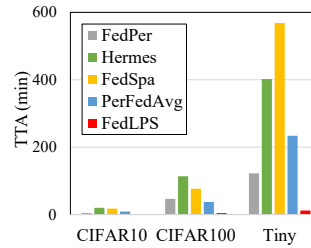


Fig. 5: TTA on CIFAR10, CIFAR100, and Tiny-Imagenet.

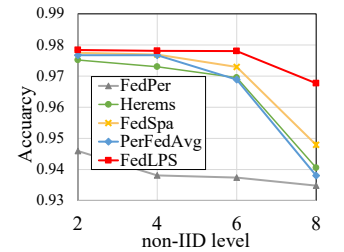


Fig. 6: Accuracy versus non-IID levels on MNIST.

curacy under the same computation costs. Noticeably, within 1500×10^{12} FLOPs, the accuracy of FedLPS in the last three rounds over CIFAR100 is 59.86%, while PerFedAvg and FedSpa provide 47.46% and 29.27% test accuracy. Besides, we observe that FedLPS can quickly converge to a target accuracy with less time than other methods, as shown in Figure 4.

We adopt Time-To-Accuracy (TTA) to represent the running time required to reach target accuracy, motivated by [29]. As shown in Figure 5, FedLPS consistently takes the shortest time to reach the target accuracy. For CIFAR10, FedLPS achieves 70% test accuracy after 90.74s, which reduces more than 68% (vs. 290.83s) running time compared to baselines. For

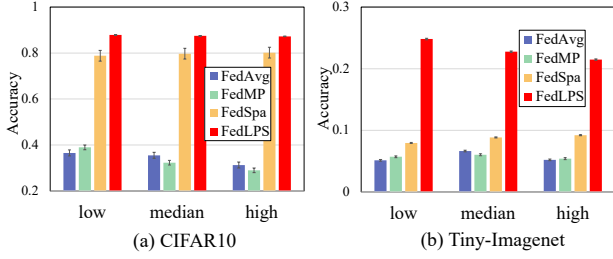


Fig. 7: Test accuracy under different system heterogeneity levels on CIFAR10 and Tiny-Imagenet.

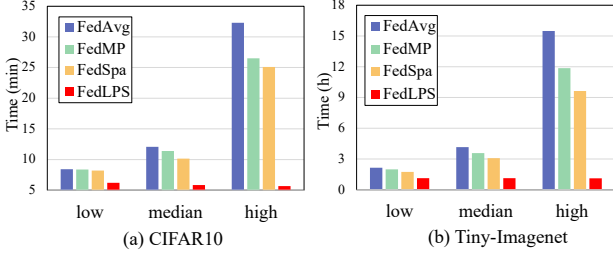


Fig. 8: Time results under different system heterogeneity levels on CIFAR10 and Tiny-Imagenet.

CIFAR100, FedLPS reaches 40% accuracy by 281.81s, which provides more than 80% (vs. 2261.89s) time saving.

D. Effect of Heterogeneity (Q3)

We also conduct experiments to evaluate FedLPS and several personalized baselines under different non-IID levels. The results on MNIST are shown in Figure 6, where the horizontal axis x implies that each client lacks x classes training data and the larger x indicates the higher non-IID level. As the non-IID level rises, the average test accuracy of different methods gradually declines. Apparently, FedLPS always outperforms baselines under different non-IID levels and the accuracy advantages are more significant on higher non-IID levels.

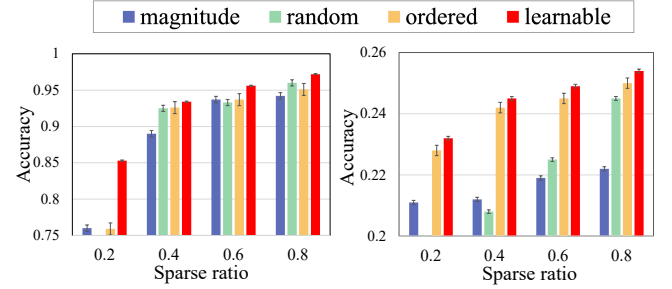
Furthermore, the effect of various heterogeneity levels is also explored. We consider three heterogeneity levels: low ($z_k \in \{1, 1/2\}$), median ($z_k \in \{1, 1/2, 1/4\}$), and high ($z_k \in \{1, 1/2, 1/4, 1/8, 1/16\}$). As shown in Figure 7 and Figure 8, FedLPS consistently keeps higher accuracy and greater time advantage, which enhances accuracy by 7.08%-55.95% and 12.28%-19.69% in CIFAR10 and Tiny-Imagenet compared to the other methods. From low to high, the running time of three baselines increases accordingly, while FedLPS can basically remain stable and shorten 24.46%-88.46% running time, demonstrating the reliability and efficiency of FedLPS.

E. Ablation Study (Q4)

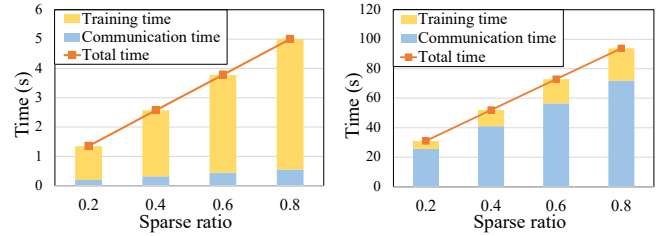
We further investigate the effect of learnable personalized patterns on a fixed sparse ratio and verify the advantage of adaptive ratios decided by P-UCBV. First, we set a fixed ratio $s_k = 0.5$ for all clients and learn personalized sparse patterns through our Learnable Sparse Training (*i.e.*, FLST). The results in Table II show that FLST generally offers higher accuracy than the start-of-the-art sparse FL works CS and

TABLE II: The results of ablation experiments.

Methods	MNIST		CIFAR10		Reddit	
	Acc (%)	FLOPs (1e12)	Acc (%)	FLOPs (1e12)	Acc (%)	FLOPs (1e12)
FLST	94.76 $\pm$ 0.06	1.4	87.32 $\pm$ 0.26	412.9	24.85 $\pm$ 0.07	5.3
RCR-Fix	94.95 $\pm$ 0.20	1.1	78.84 $\pm$ 0.10	394.3	23.39 $\pm$ 0.56	5.1
P-UCBV-Fix	95.72$\pm$0.26	0.9	80.71$\pm$0.44	325.0	23.71$\pm$0.08	3.7
RCR-Dyn	95.99 $\pm$ 0.12	1.1	86.06 $\pm$ 0.12	380.1	25.20 $\pm$ 0.06	4.6
P-UCBV-Dyn	96.77$\pm$0.11	0.8	87.43$\pm$0.19	268.5	26.17$\pm$0.04	4.2



(a) Test accuracy versus sparse ratio by different sparsification strategies on MNIST (left) and Reddit (right).



(b) Time results of our learnable sparsification on MNIST (left) and Reddit.

Fig. 9: Accuracy and time results under different sparse ratios.

FedSpa (see Table I, where CS and FedSpa also adopt the same ratio $s_k = 0.5$) under the same FLOPs, revealing the vital contribution of our learnable personalized pattern. To quantitatively verify the advantages of learnable sparsification, we compare against existing heuristic sparsifications (including random, ordered, and magnitude-based strategies) under different sparse ratios, where the sparse ratio is set to $s_k \in \{0.2, 0.4, 0.6, 0.8\}$ for all clients. As shown in Figure 9a, FedLPS consistently outperforms other methods under various ratio settings, demonstrating the reliability of our learnable strategy. Besides, we observe from Figure 9b that as the ratio increases, the test accuracy is enhanced while running time also increases, indicating the significance of the ratio selection.

To verify the advantages of P-UCBV in ratio decisions, we compare it with a rigid Resourced-Controlled Ratio (RCR) rule under fixed and dynamic system heterogeneity levels. Sparse ratios are directly set according to local computation capabilities in RCR, which is used in [27, 37], and [31]. As shown in Table II, RCR shows lower accuracy and more FLOPs than P-UCBV, which indicates that our P-UCBV significantly reduces training costs while guaranteeing inference accuracy.

VI. RELATED WORK

Model Sparsification. Modern DNNs typically contain millions of parameters, requiring high computing power for training [63]. For resource-limited edge devices, it is infeasible to completely train dense models. Hence, for lightening models, sparsification has attracted widespread attention [64–67], which can be characterized as unstructured and structured. *Unstructured sparsification* simply zeros out partial parameters without considering structures of DNNs, where sparse parameter matrices are too irregular to accelerate training on commodity hardware [68]. Although several works [38, 69] apply unstructured sparsification into FL, they require specialized hardware/libraries on clients and cannot adapt to diverse configurations [28]. On the contrary, *structured sparsification* [68, 70] takes DNN structures (*i.e.*, units) into account and implements structure-wise pruning. The remaining part after structured sparsification is regular and can be viewed as a submodel, thereby enabling training speed-ups on general-purpose hardware. Federated Dropout [36] (FD) first introduces structured sparsification into FL for MEC, but it adopts uniform sparse ratios and random sparse patterns, ignoring ratio determinations and incurring precision sacrifices.

System Heterogeneity. System heterogeneity is one of the primary challenges of edge data management and computing. Conventional FL requires global and local models to share the same architecture [71], where low-tier clients are excluded from training, leading to training bias and accuracy degradation [72]. Knowledge distillation [73, 74] supports training different models, but requires public datasets for fine-tuning, which violates the data localization and adds extra overhead [31]. Several works explore sparsification to extract different submodels. The common idea is to assign sparse models with different sizes based on local resources, where sparse ratios determine model sizes. FedDrop [32] adjusts sparse ratios depending on bandwidth limitation, while Fjord [37] selects ratios based on local computing power. They focus on the association between sparse ratios and resources without involving local data, resulting in unguaranteed performance in non-IID settings. For sparse pattern selection, existing works can be roughly categorized into depth and width scaling [75]. Depth scaling [58, 76] constructs heterogeneous models by removing several deepest layers. Such layer-wise sparsification cannot support fine-grained adjustments of patterns and is more prone to larger accuracy fluctuations. In contrast, width scaling tailors sparse models by pruning neurons/channels. Fjord [37], HeteroFL [31], and FedRolex [27] tailor sparse models in an ordered manner, meaning that adjacent units are dropped out first in these works. While FedMP [28] prunes the units with lower magnitude. The above ordered and magnitude-based sparsifications are strongly heuristic [77] and cannot flexibly customize sparse patterns. Finally, a shared model is obtained for inference on all clients, failing to generalize well on clients with non-IID data [78].

Sparsification with Statistical Heterogeneity. The non-IID

problem is common in EDM and MEC due to the distinct preferences of edge users [49]. To tackle the problem, several FL works propose to embed personalized information into client-side sparse models. Hermes [33] and LotteryFL [42] heuristically prune lower-magnitude parameters or units to tailor a personalized sparse model for each client. However, the magnitude cannot accurately reflect the importance of model units over local data, resulting in accuracy degradation. Moreover, they take dense-to-sparse rules, gradually decreasing sparse ratios at a fixed rate after reaching preset accuracy thresholds. In this way, weaker clients are still required to train larger scaling submodels and suffer from unbearable training loads, which delays the FL progress and even induces stragglers. Sparse-to-sparse techniques are explored in FedSpa [34], where each client shifts the submodel regularly by the iterative cuts of the lowest magnitude parameters and random growth of other connections during local training, where heuristic magnitude-based pattern adjustment also cannot guarantee accuracy. Besides, FedSpa restricts clients to hold the same and constant sparse ratio, unable to be applied in system-heterogeneous MEC scenarios. Although the state-of-the-art FedP3 [35] considers system heterogeneity, it still adopts heuristic sparse patterns (*i.e.*, uniform and ordered dropout).

VII. CONCLUSION

There has been a growing interest in distributed data management at network edges to utilize data in a real-time and privacy-preserving way. In this paper, we focus on the challenges of statistical and system heterogeneity in such edge scenarios, and propose Learnable Personalized Sparsification for heterogeneous Federated learning (FedLPS), which facilitates the learnable customization of sparse patterns and sparse ratios to boost model performance and computing efficiency. We integrate the importance of model units into local loss and learn the importance on local data to tailor importance-based sparse patterns with minimal heuristics, which can accurately extract personalized data features. Furthermore, P-UCBV learns the additive effect of diverse capabilities and non-IID data via the MAB modeling and superimposed feedback designing to adaptively determine sparse ratios for the trade-off between capability adaption and accuracy improvement. We conduct extensive experiments on typical datasets with various models, and the results show that FedLPS significantly outperforms the state-of-the-art method in test accuracy and training efficiency.

Our learnable idea inspires that pending indicators might be learned together with models in a non-heuristic way, which provides a future direction for learnable hyperparameter customization in edge data management and computing.

ACKNOWLEDGEMENTS

This article is supported in part by the National Key Research and Development Program of China (No. 2021YFB2900102); in part by the National Natural Science Foundation of China (NSFC) under Grants 62472410 and 62072436. Additionally, Jingyuan Wang's work was partially supported by NSFC (No. 72171013, 72222022, and 72242101).

REFERENCES

- [1] Z. Liu, J. Wang, Z. Li, and Y. He, "Full bayesian significance testing for neural networks in traffic forecasting," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.
- [2] J. Wang, N. Wu, and W. X. Zhao, "Personalized route recommendation with neural network enhanced search algorithm," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5910–5924, 2022.
- [3] D. O’Keeffe, T. Salonidis, and P. Pietzuch, "Frontier: resilient edge processing for the internet of things," *Proc. VLDB Endow.*, vol. 11, no. 10, p. 1178–1191, 2018.
- [4] I. Lujic, V. De Maio, and I. Brandic, "Resilient edge data management framework," *IEEE Transactions on Services Computing*, vol. 13, no. 4, pp. 663–674, 2020.
- [5] J. Jiang, D. Pan, H. Ren, X. Jiang, C. Li, and J. Wang, "Self-supervised trajectory representation learning with temporal regularities and travel semantics," in *IEEE 39th International Conference on Data Engineering (ICDE)*, 2023.
- [6] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, and P. Hui, "Edge intelligence: Empowering intelligence to the edge of network," *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1778–1837, 2021.
- [7] J. Liu, Y. Xu, H. Xu, Y. Liao, Z. Wang, and H. Huang, "Enhancing federated learning with intelligent model migration in heterogeneous edge computing," in *IEEE 38th International Conference on Data Engineering (ICDE)*, 2022.
- [8] Y. Gong, Y. Li, and N. M. Freris, "FedADMM: A robust federated deep learning framework with adaptivity to system heterogeneity," in *IEEE 38th International Conference on Data Engineering (ICDE)*, 2022.
- [9] M. Hu, P. Zhou, Z. Yue, Z. Ling, Y. Huang, Y. Liu, and M. Chen, "FedCross: Towards accurate federated learning via multi-model cross aggregation," in *IEEE 40th International Conference on Data Engineering (ICDE)*, 2024.
- [10] R. R. Pansara, "Edge computing in master data management: Enhancing data processing at the source," *International Transactions in Artificial Intelligence*, vol. 6, no. 6, p. 1–11, 2022.
- [11] S. Jere, Q. Fan, B. Shang, L. Li, and L. Liu, "Federated learning in mobile edge computing: An edge-learning perspective for beyond 5g," *arXiv preprint arXiv:2007.08030*, 2020.
- [12] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, 2023.
- [13] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey," *Neurocomputing*, vol. 597, p. 128019, 2024.
- [14] J. Mirval, L. Bouganim, and I. Sandu Popa, "Federated learning on personal data management systems: Decentralized and reliable secure aggregation protocols," in *Proceedings of the 35th International Conference on Scientific and Statistical Database Management*, 2023.
- [15] Y. Cheng, L. Zhang, J. Wang, X. Chu, D. Huang, and L. Xu, "FedMix: Boosting with data mixture for vertical federated learning," in *IEEE 40th International Conference on Data Engineering (ICDE)*, 2024, pp. 3379–3392.
- [16] J. Pei, W. Liu, J. Li, L. Wang, and C. Liu, "A review of federated learning methods in heterogeneous scenarios," *IEEE Transactions on Consumer Electronics*, 2024.
- [17] J. Liu, Y. Zhou, D. Wu, M. Hu, M. Guizani, and Q. Z. Sheng, "FedLMT: Tackling system heterogeneity of federated learning via low-rank model training with theoretical guarantees," in *International Conference on Machine Learning (ICML)*, 2024.
- [18] Z. Lu, H. Pan, Y. Dai, X. Si, and Y. Zhang, "Federated learning with non-iid data: A survey," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19 188–19 209, 2024.
- [19] Z. Qin, S. Deng, M. Zhao, and X. Yan, "FedAPEN: Personalized cross-silo federated learning with adaptability to statistical heterogeneity," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.
- [20] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *International Conference on Machine Learning (ICML)*, 2020.
- [21] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International Conference on Machine Learning (ICML)*, 2021.
- [22] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "FedCP: Separating feature information for personalized federated learning via conditional policy," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.
- [23] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2021.
- [24] A. M. Abdelmoniem, A. N. Sahu, M. Canini, and S. A. Fahmy, "REFL: Resource-efficient federated learning," in *Proceedings of the Eighteenth European Conference on Computer Systems*, 2023.
- [25] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine learning and systems (MLSys)*, vol. 2, 2020, pp. 429–450.
- [26] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, 2021.
- [27] S. Alam, L. Liu, M. Yan, and M. Zhang, "FedRolex: Model-heterogeneous federated learning with rolling sub-model extraction," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [28] Z. Jiang, Y. Xu, H. Xu, Z. Wang, C. Qiao, and Y. Zhao, "FedMP: Federated learning through adaptive model pruning in heterogeneous edge computing," in *IEEE 38th International Conference on Data Engineering (ICDE)*, 2022.
- [29] J. Xue, M. Liu, S. Sun, Y. Wang, H. Jiang, and X. Jiang, "FedBIAD: Communication-efficient and accuracy-guaranteed federated learning with bayesian inference-based adaptive dropout," in *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2023.
- [30] Z. Chen, C. Jia, M. Hu, X. Xie, A. Li, and M. Chen, "FlexFL: Heterogeneous federated learning via apoz-guided flexible pruning in uncertain scenarios," *arXiv preprint arXiv:2407.12729*, 2024.
- [31] E. Diao, J. Ding, and V. Tarokh, "HeteroFL: Computation and communication efficient federated learning for heterogeneous clients," in *International Conference on Learning Representations (ICLR)*, 2021.
- [32] D. Wen, K.-J. Jeon, and K. Huang, "Federated dropout—a simple approach for enabling federated learning on resource constrained devices," *IEEE Wireless Communications Letters*, vol. 11, no. 5, pp. 923–927, 2022.
- [33] A. Li, J. Sun, P. Li, Y. Pu, H. Li, and Y. Chen, "Hermes: An efficient federated learning framework for heterogeneous mobile clients," in *International Conference on Mobile Computing and Networking*, 2021.
- [34] T. Huang, S. Liu, L. Shen, F. He, W. Lin, and D. Tao, "Achieving personalized federated learning with sparse local models," *arXiv preprint arXiv:2201.11380*, 2022.
- [35] K. Yi, N. Gazagnadou, P. Richtárik, and L. Lyu, "FedP3: Federated personalized and privacy-friendly network pruning under model heterogeneity," in *International Conference on Learning Representations (ICLR)*, 2024.
- [36] S. Caldas, J. Konečný, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.
- [37] S. Horvath, S. Laskaridis, M. Almeida, I. Leontiadis, S. Venieris, and N. Lane, "Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [38] X. Jiang and C. Borcea, "Complement sparsification: Low-overhead model pruning for federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.
- [39] N. Bouacida, J. Hou, H. Zang, and X. Liu, "Adaptive federated dropout: Improving communication efficiency and generalization for federated learning," *IEEE INFOCOM - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2021.
- [40] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning (ICML)*, 2021.
- [41] X. Li, M. Liu, S. Sun, Y. Wang, H. Jiang, and X. Jiang, "FedTrip: A resource-efficient federated learning method with triplet regularization," in *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2023.
- [42] 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 *IEEE/ACM Symposium on Edge Computing (SEC)*, 2021.
- [43] S. Mukherjee, K. Naveen, N. Sudarsanam, and B. Ravindran, "Efficient-UCBV: An almost optimal algorithm using variance estimates," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [44] S. Q. Zhang, J. Lin, and Q. Zhang, "A multi-agent reinforcement

- learning approach for efficient client selection in federated learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.
- [45] R. Dai, L. Shen, F. He, X. Tian, and D. Tao, “DisPFL: Towards communication-efficient personalized federated learning via decentralized sparse training,” in *International Conference on Machine Learning (ICML)*, 2022.
- [46] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor, “Federated learning with differential privacy: Algorithms and performance analysis,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.
- [47] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, “BatchCrypt: Efficient homomorphic encryption for Cross-Silo federated learning,” in *USENIX Annual Technical Conference (USENIX ATC)*, 2020.
- [48] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.
- [49] Y. Wang, Y. Tong, Z. Zhou, R. Zhang, S. J. Pan, L. Fan, and Q. Yang, “Distribution-regularized federated learning on non-iid data,” in *IEEE 39th International Conference on Data Engineering (ICDE)*, 2023.
- [50] 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 Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2022.
- [51] M. Chen, Y. Xu, H. Xu, and L. Huang, “Enhancing decentralized federated learning for non-iid data on heterogeneous devices,” in *IEEE 39th International Conference on Data Engineering (ICDE)*, 2023.
- [52] J. Xu, S. Wang, L. Wang, and A. C.-C. Yao, “FedCM: Federated learning with client-level momentum,” *arXiv preprint arXiv:2106.10874*, 2021.
- [53] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” 2010. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [54] A. Krizhevsky, G. Hinton *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [55] Y. Le and X. Yang, “Tiny imagenet visual recognition challenge,” *CS 231N*, vol. 7, no. 7, p. 3, 2015.
- [56] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, “LEAF: A benchmark for federated settings,” *arXiv preprint arXiv:1812.01097*, 2019.
- [57] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [58] M. Kim, S. Yu, S. Kim, and S.-M. Moon, “DepthFL: Depthwise federated learning for heterogeneous clients,” in *International Conference on Learning Representations (ICLR)*, 2023.
- [59] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, “Federated learning with personalization layers,” *arXiv preprint arXiv:1912.00818*, 2019.
- [60] A. Fallah, A. Mokhtari, and A. Ozdaglar, “Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [61] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. Van Gool, “Ai benchmark: Running deep neural networks on android smartphones,” in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.
- [62] H. Bouzidi, H. Ouarnoughi, S. Niar, and A. A. E. Cadi, “Performance prediction for convolutional neural networks in edge devices,” *arXiv preprint arXiv:2010.11297*, 2020.
- [63] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, “Model compression and acceleration for deep neural networks: The principles, progress, and challenges,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126–136, 2018.
- [64] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, “Rethinking the value of network pruning,” in *International Conference on Learning Representations (ICLR)*, 2019.
- [65] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen, “Rigging the lottery: Making all tickets winners,” in *International Conference on Machine Learning (ICML)*, 2020.
- [66] Y. Zhang, Y. Yao, P. Ram, P. Zhao, T. Chen, M. Hong, Y. Wang, and S. Liu, “Advancing model pruning via bi-level optimization,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [67] C. Tianlong, Z. A. Zhang, A. Jaiswal, S. Liu, and Z. Wang, “Sparse moe as the new dropout: Scaling dense and self-slimmable transformers,” *arXiv preprint arXiv:2303.01610*, 2023.
- [68] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, “Learning structured sparsity in deep neural networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [69] S. Bibikar, H. Vikalo, Z. Wang, and X. Chen, “Federated dynamic sparse training: Computing less, communicating less, yet learning better,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.
- [70] A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T.-J. Yang, and E. Choi, “MorphNet: Fast and simple resource-constrained structure learning of deep networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [71] T. Nishio and R. Yonetani, “Client selection for federated learning with heterogeneous resources in mobile edge,” *IEEE International Conference on Communications (ICC)*, 2019.
- [72] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, and et.al., “Advances and open problems in federated learning,” *arXiv preprint arXiv:1912.04977*, 2019.
- [73] D. Li and J. Wang, “FedMD: Heterogeneous federated learning via model distillation,” *arXiv preprint arXiv:1910.03581*, 2019.
- [74] Z. Zhu, J. Hong, and J. Zhou, “Data-free knowledge distillation for heterogeneous federated learning,” in *International Conference on Machine Learning (ICML)*, 2021.
- [75] C. Jia, M. Hu, Z. Chen, Y. Yang, X. Xie, Y. Liu, and M. Chen, “AdaptiveFL: Adaptive heterogeneous federated learning for resource-constrained aiot systems,” in *Design Automation Conference (DAC)*, 2024.
- [76] R. Liu, F. Wu, C. Wu, Y. Wang, L. Lyu, H. Chen, and X. Xie, “No one left behind: Inclusive federated learning over heterogeneous devices,” in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.
- [77] A. Kusupati, V. Ramanujan, R. Somani, M. Wortsman, P. Jain, S. Kakade, and A. Farhadi, “Soft threshold weight reparameterization for learnable sparsity,” in *International Conference on Machine Learning (ICML)*, 2020.
- [78] A. Z. Tan, H. Yu, L. zhen Cui, and Q. Yang, “Towards personalized federated learning,” *IEEE transactions on neural networks and learning systems*, pp. 1–17, 2021.